THE UNIVERSITY *of* EDINBURGH

Robustness and Explainability in Automatic Speech Recognition and Speaker Verification

Xiaoliang Wu

Doctor of Philosophy

THE UNIVERSITY OF EDINBURGH

2025

To my family.

Abstract

In recent years, speech-based artificial intelligence (AI) systems—such as Automatic Speech Recognition (ASR) and Speaker Verification (SV)—have achieved remarkable performance and are now widely used in applications such as virtual assistants, security authentication, and healthcare. This progress has been largely driven by deep learning models, which offer powerful representation learning and have substantially boosted system accuracy. However, their black – box nature has also introduced new challenges. As these systems become increasingly integrated into real-world scenarios, concerns have emerged regarding their robustness – how well they perform under adversarial or unexpected conditions—and their explainability—whether their outputs can be meaningfully interpreted and trusted. These concerns are not only of practical relevance but are also reflected in emerging regulatory frameworks, such as the EU AI Act, which emphasize the need for AI systems to be both robust and explainable. While explainability research has advanced significantly in computer vision and natural language processing (NLP), the speech domain remains comparatively underexplored. This thesis addresses this gap by investigating both robustness and explainability in the context of ASR and SV.

The first aspect we investigate is robustness. A robust model should produce consistent outputs under small, imperceptible changes to the input. To examine this property in ASR systems, we construct adversarial perturbations that sound nearly the same to human listeners but result in different transcriptions. We make use of a property of human hearing, where louder sounds in a frequency band can mask softer ones occurring at the same time. By placing perturbations under these stronger parts of the signal, we create audio that is indistinguishable to humans but still misleads the ASR system. This approach exposes the fragility of ASR models under subtle changes that are inaudible to humans and serves as a tool to evaluate their robustness.

Having examined robustness, we next focus on explainability. We begin by proposing XASR, a modular and explainable ASR architecture that integrates multiple post-hoc explanation methods – that is, techniques applied after model inference to interpret its predictions – such as LIME, SFL, and Causal – to analyze model predictions at the input level. These methods aim to highlight which parts of the input signal, such as specific time frames, are most responsible for the system's output. However, there is no standard ground truth for what constitutes a correct explanation. As a result, many evaluations rely on user studies, which are subjective and difficult to reproduce. This limits our ability to assess the validity of post-hoc methods.

To examine the validity of post-hoc explanations on ASR, we use a phoneme recognition model built with Kaldi and trained on the TIMIT dataset. This setup offers ground-truth frame-level phoneme alignments and a simple, controllable architecture, making it ideal for systematic analysis. We apply standard LIME and propose two speech-specific variants – LIME-WS (Window Segment) and LIME-TS (Time Segment) –which restrict perturbations to a narrow temporal window around the phoneme of interest. This localized focus better reflects the temporal nature of speech. By comparing these outputs to ground-truth phoneme boundaries, we quantitatively assess whether the methods accurately highlight the input regions responsible for specific predictions. However, such methods only analyze the input-output relationship from the outside, without revealing how the model actually processes the input internally. This limitation motivates a shift toward intrinsic explainability – that is, designing models whose internal representations are interpretable by construction, without relying on external post-hoc methods. We explore this idea by building explainable representations within the model itself.

To achieve this goal, we turn to speaker verification, which offers a more natural setting for exploring intrinsic explainability. Unlike ASR, which relies on detailed frame-level acoustic modeling, SV relies more on high-level speaker traits. This makes it more natural to connect model behavior with human-understandable attributes, such as accent, nationality, or profession. We propose a concept-based SV model in which each intermediate dimension corresponds to a human-understandable attribute supervised using annotated metadata. This structure allows the model to express verification decisions in explainable terms, such as identifying speaker similarity based on shared accent.

Through these contributions—designing adversarial frequency masking attacks, developing the XASR architecture with post-hoc explanation integration, validating post-hoc methods with controlled phoneme-level settings, and introducing a concept-based intrinsic explainability model for SV—this thesis lays a foundation for developing speech systems that are not only effective, but also robust, explainable, compliant with emerging regulations, and more likely to earn user trust.

Lay Summary

Speech-based artificial intelligence (AI) systems—like those that let you talk to virtual assistants or unlock your phone with your voice—are becoming more common in everyday life. These systems are also starting to play a role in sensitive areas such as healthcare. But as they take on more important tasks, two key concerns have emerged:

1. **Robustness** – do these systems still work well when the sound changes just a little, such as in the presence of noise?
2. **Explainability** – can we understand why the system gave a certain answer?

To address the first concern, we explore how speech recognition systems react to tiny changes in the audio—changes that humans can't hear but that cause the system to produce different results. By creating examples of these confusing situations, we help reveal where the systems are vulnerable and provide insight into how we can design models that are more stable and dependable.

For the second concern, we study how to make the decision-making process of speech AI more transparent. We investigate methods that highlight what parts of the audio the system focuses on when making a prediction. We also test how well these explanations match what the system is actually doing, using a carefully controlled setting. Finally, we design a speech-based system that includes human understandable factors – such as a person's accent – so that its decisions can be explained in familiar terms. This is particularly helpful in voice identity tasks, where knowing why two voices are judged similar or different is important for building trust and catching errors.

This research contributes to building voice-based AI systems that are not only accurate, but also safer and easier for people to understand. It helps ensure that future speech technologies can be used with greater confidence in real-world settings, where reliability and transparency matter most.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisors, Prof. Ajitha Rajan and Prof. Peter Bell, for their invaluable guidance, support, and encouragement throughout my PhD journey. Their insights, patience, and belief in my work have shaped this thesis and my development as a researcher.

I am deeply thankful to my family for their unwavering love and support. Their quiet strength and constant encouragement have been my anchor through every stage of this journey.

To my dear friends Xuran, Yanfei, Yanjing, Na and Huanhuan—thank you for being there through the ups and downs, for your companionship, and for reminding me that life is more than just research.

I am also grateful to my collaborators Chau Luu, Ke Liu and Yi Wang for the stimulating discussions and generous sharing of ideas. Working with you has been both productive and inspiring.

Lastly, I want to thank myself—for never backing down, even in moments of deep self-doubt and self-denial. May I continue to move forward, with steady steps and unwavering faith.

To all those who have supported me, challenged me, or simply listened along the way—thank you.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

Xiaoliang Wu

Contents

Figures and Tables

Figures

Tables

Chapter 1

Introduction

In recent years, speech-based artificial intelligence (AI) has seen widespread deployment across consumer, commercial, and public service domains, becoming a key modality for human-computer interaction. Among its many components, automatic speech recognition (ASR) and speaker verification (SV) have emerged as two of the most widely adopted technologies.

ASR enables machines to transcribe and interpret spoken language, and now supports a variety of everyday applications. These include virtual assistants such as Amazon Alexa [5] and Apple Siri [6], real-time transcription services in platforms like Zoom Live Captions [7] and Google Meet [8], and dictation tools on iOS and Android devices [9]. SV systems, by contrast, verify a speaker's identity from their voice and are increasingly deployed for biometric authentication. HSBC, for example, has used voice identification in telephone banking since 2016, with over 15 million users reported by 2022 [10]. Similar capabilities are offered by Microsoft Azure's speaker recognition API [11] and Google Voice Match [12], supporting personalized access on shared smart devices. Beyond consumer services, both ASR and SV have been adopted in high-stakes public domains, including emergency call triage [13], forensic speaker comparison [14], and preliminary health assessments using vocal biomarkers [15, 16].

As ASR and SV technologies are applied in more critical and sensitive domains, questions about their reliability with increasing reliance on AI, transparency, and accountability have become increasingly relevant. In response, regulatory frameworks have emerged to ensure that such systems meet appropriate standards of reliability, transparency, and safety. Most notably, the European Union's Artificial Intelligence Act (EU AI Act) [17] introduces legally binding requirements for a broad class of "high-risk AI systems," including those involving biometric identification, behavioral inference, and voice-based interaction. Many speech-based applications—including ASR, SV, and speaker profiling—fall under this scope. According to

the Act, such systems must demonstrate adequate levels of accuracy and robustness, while also ensuring that they are transparent enough for users to interpret and use appropriately, and that appropriate human oversight is in place[1]. Similar expectations are reflected in the OECD AI Principles and the U.S. NIST AI Risk Management Framework [18, 19].

Together, these developments reflect a critical shift: strong performance alone is no longer sufficient. For AI systems to be trustworthy and legally compliant, they must also be robust to variation and capable of providing human-understandable explanations. These two technical properties—robustness and explainability—are now widely recognized as essential conditions for the safe and responsible deployment of speech AI systems.

The first issue concerns robustness. In practical settings, a reliable system is expected to produce consistent outputs despite small, semantically irrelevant changes to the input—such as variations in background noise, recording devices, or speaking environments. However, recent studies have shown that ASR models may exhibit unstable behavior under such conditions. Shah et al. [20] demonstrated that small, imperceptible perturbations can lead ASR systems to produce entirely different transcriptions. Similarly, Cissé et al. found that adding typical telephony noise, which does not alter human perception, can nonetheless degrade recognition quality [21]. These findings indicate that, despite their high performance in controlled environments, many speech models lack the robustness required for deployment in real-world scenarios.

Such limitations are not merely theoretical but have practical implications in real-world deployments. In safety-critical contexts, they may lead to serious consequences. For instance, Yuan et al. [22] demonstrated that adversarial commands embedded in background music—termed "CommanderSong"—could be used to manipulate smart assistants without the user's knowledge. In another case, the failure of ASR to correctly transcribe speech during emergency calls has been identified as a contributing factor to delayed response times and misrouted services [23]. These examples highlight that insufficient robustness not only reduces reliability but also introduces security risks and the potential for harm.

A second major concern is the lack of explainability—the capacity of a system to provide human-understandable reasons for its outputs. As speech AI systems are increasingly applied in sensitive decision-making scenarios, users and stakeholders must be able to comprehend how these systems function and what aspects of the input influence their predictions. However, in practice, such explanations are often unavailable. For instance, a 2022 audit conducted by the Mozilla Foundation documented significant disparities in ASR accuracy across speaker accents, age groups, and gender identities, yet none of the evaluated systems provided any insight into which components of the speech signal contributed to these differences [24]. In the context of speaker verification, The Guardian reported in 2023 that a visually impaired

1. Articles 13–15 of the EU Artificial Intelligence Act (Regulation (EU) 2024/1689) specify the requirements for transparency (Art. 13), human oversight (Art. 14), and accuracy, robustness and explainability (Art. 15).

individual was repeatedly denied access by a voice authentication system, with no indication of which vocal features were used or why the match failed [25]. In both cases, the absence of explainable output limits users' ability to understand, contest, or meaningfully interact with the system's decisions.

To address the challenges raised above, this thesis investigates these two central challenges: robustness and explainability. The first part focuses on robustness in the context of ASR, where we design black-box adversarial attacks to reveal how small, human-imperceptible perturbations can lead to significant transcription errors. These attacks do not require access to model internals, making them suitable for evaluating real-world systems. The second part turns to the explainability, which is explored in both ASR and SV. In ASR, we apply post-hoc explainable methods[2] to analyze which parts of the input influence recognition outcomes. In SV, we design intrinsically interpretable models[3] based on human-understandable speaker attributes.

The following sections describe each of these directions in detail.

1.1 Robustness of ASR under Adversarial Perturbations

This part of the thesis investigates the robustness of ASR systems by exposing their vulnerabilities through adversarial audio.

Previous research has shown that even minor, human-imperceptible changes to input audio can cause deep learning-based ASR systems to generate incorrect transcriptions [3]. However, most existing attack methods rely on white-box assumptions, requiring access to gradients or internal model parameters [26]. These constraints make them unsuitable for commercial or closed-source systems, and many of these methods either introduce audible artifacts or involve high computational costs. Although black-box attacks have been proposed [2, 27], they often depend on repeated transcription queries and remain limited in flexibility or efficiency.

To address these limitations and better probe the vulnerabilities of real-world ASR systems, we propose SPAT (Speech Psychoacoustic Adversarial Tool), a black-box and untargeted attack framework that requires no access to model internals or transcription outputs. SPAT applies perturbations in perceptually insignificant frequency regions based on psychoacoustic masking, thereby preserving audio quality. We further introduce a frame selection strategy that prioritizes high-impact regions for perturbation, improving both attack efficiency and effectiveness.

---

2. Post-hoc methods refer to techniques applied after model inference to interpret the decision, such as identifying which parts of the input were most influential.

3. Intrinsic explainability involves building models whose internal representations are aligned with human-understandable concepts, such as speaker attributes (e.g., accent, gender, or profession).

We evaluate SPAT on three ASR systems—DeepSpeech [28], Sphinx [29], and Google Cloud Speech-to-Text [30]—using two standard datasets. Compared to existing blackbox attacks, SPAT achieves higher success rates, requires fewer resources, and is less likely to be detected by known defense systems [4].

1.2 Explainability in ASR and SV

The second component of this thesis is to improve the explainability of ASR and SV. Compared to computer vision and natural language processing, explainability in speech remains limited. Existing methods are mostly applied to speech classification tasks such as speaker identification, emotion recognition, or accent detection [31, 32, 33]. These approaches often use techniques to highlight which parts of the input signal—such as specific time frames or frequency regions—contribute to the final output. For sequence-to-sequence tasks like ASR, however, explanation methods remain few, and their reliability is not well understood.

To address this gap, we propose XASR, a unified and model-agnostic framework for post-hoc explanation in ASR. XASR integrates multiple explanation methods, including Local Interpretable Model-agnostic Explanations (LIME) [34], Statistical Fault Localization (SFL) [35] and Causal [36], to generate frame-level importance scores that highlight which portions of the input signal are responsible for a given transcription. This framework operates without access to model internals, making it applicable to both open-source and commercial systems. We evaluate XASR across multiple ASRs (DeepSpeech [28], Sphinx [29], Google [30]) and use it to identify input segments associated with recognition errors.

However, a core limitation of post-hoc explanation methods is the lack of ground truth for what constitutes a correct explanation [37]. Most evaluations rely on user studies, which are subjective and difficult to reproduce. To address this, we conduct a controlled analysis using a phoneme recognition model built with Kaldi [38] and trained on the TIMIT dataset [39]. This setup includes ground-truth frame-level phoneme alignments and a simplified architecture suitable for systematic testing. Within this setting, we propose two speech-specific variants of LIME—LIME-WS (Window Segment) and LIME-TS (Time Segment)—that limit perturbations to time intervals around each phoneme. We compare the output of these methods to phoneme boundaries to assess whether the identified regions align with linguistically meaningful segments. This allows for quantitative evaluation of post-hoc explanations in ASR tasks.

However, post-hoc methods only analyze the input-output relationship but do not reveal how the model processes information internally. Based on this limitation, we shift focus to intrinsic explainability, where the model is structured to reflect human-understandable concepts. We explore this in the context of speaker verification, where decisions often depend on high-level speaker attributes such as accent, nationality, or profession. We design a model that takes

these attributes as input and predicts whether two utterances belong to the same speaker. During training, each attribute is explicitly supervised using available metadata, ensuring that the model learns to organize its decision process around interpretable factors. At inference time, we can examine which attributes led to a verification decision by observing which inputs triggered a match or mismatch, without relying on post-hoc explanation methods.

These three studies address key gaps in explainability for ASR and SV. The first proposes a general framework for generating input-level explanations for ASR outputs without access to model internals. The second introduces evaluation methods to assess the reliability of these explanations using controlled experiments. The third shifts to intrinsic explainability by designing a speaker verification model that uses human-interpretable attributes to make predictions. Together, they form a progression from external analysis to interpretable model design.

From the above discussion, three overarching research questions (RQs) are distilled in order to make the motivations clearer and to facilitate the reader's understanding. These RQs serve as a unifying thread throughout the thesis, with each subsequent chapter contributing to answering them.

RQ1: How do Automatic Speech Recognition (ASR) systems behave under adversarial perturbation? *Addressed in Chapter 3.*

RQ2: How effective are post-hoc explainability methods when applied to speech AI, particularly ASR, and how can their reliability be systematically evaluated? *Addressed in Chapters 4 and 5.*

RQ3: Can intrinsic explainability be achieved in speaker verification (SV) by aligning intermediate model representations with human-understandable attributes, and what insights and limitations does this reveal? *Addressed in Chapter 6.*

1.3 Contributions

This section presents a structured summary of the thesis contributions, as illustrated in Figure 1.1. The figure outlines the core developments of each chapter and helps guide the reader through the overall flow of the work.

1. **Black-box adversarial attack against ASR (Chapter 3)** – We propose *SPAT*, a black-box adversarial attack framework. The method uses psychoacoustic masking and frame-level selection to generate efficient, imperceptible perturbations, revealing vulnerabilities in both open-source and commercial ASR systems.

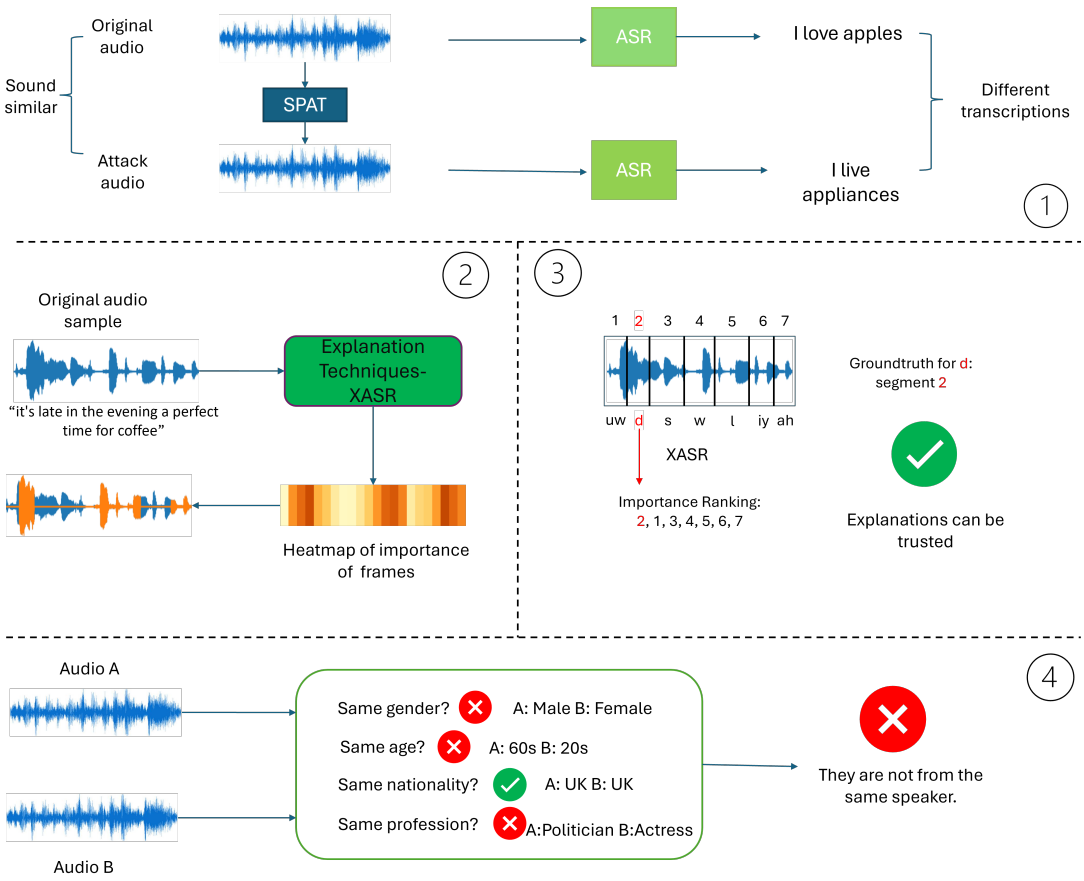


Figure 1.1: Overview of contributions.

2. **Perturbation-based post-hoc explainability in ASR (Chapter 4)** – We conclude a model-agnostic framework that combines several perturbation-based post-hoc explainable methods into a unified pipeline. We further adapt these methods for the temporal and sequential nature of speech data to improve their applicability in ASR tasks.
3. **Quantitative evaluation of explanation quality on Phoneme Recognition (Chapter 5)** – We develop an evaluation protocol for post-hoc explanations in ASR by leveraging a phoneme recognition model trained on aligned speech data.
4. **An intrinsically explainable SV model (Chapter 6)** – We propose a concept-based SV model in which intermediate representations correspond to human-interpretable speaker attributes such as accent and profession. The model is trained with attribute-level supervision, enabling verification decisions to be traced directly to concept activations rather than inferred post hoc.

1.4 Peer-Reviewed Publications

This thesis has led to the following peer-reviewed publications, which reflect its main research contributions.

1. **Xiaoliang Wu**, Peter Bell, and Ajitha Rajan. 2023a. Can We Trust Explainable AI Methods on ASR? An Evaluation on Phoneme Recognition. arXiv: 2305.18011 [cs.CL] (Accepted by ICASSP 2024)
2. **Xiaoliang Wu**, Peter Bell, and Ajitha Rajan. 2023b. “Explanations for automatic speech recogni- tion”. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2023).
3. **Xiaoliang Wu** and Ajitha Rajan. 2022. “Catch Me If You Can: Blackbox Adversarial Attacks on Automatic Speech Recognition using Frequency Masking”. In Proceedings of 29th Asia-Pacific Software Engineering Conference (APSEC).
4. **Xiaoliang Wu**. 2022. “Blackbox adversarial attacks and explanations for automatic speech recognition”. In Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 1765–1769
5. Liu, Ke ; Gao, Shangde ; **Wu, Xiaoliang** et al. / Mat-Instructions : A large-scale inorganic material instruction dataset for large language models. Proceedings of the 34th International Joint Conference on Artificial Intelligence (IJCAI-25). Institute of Electrical and Electronics Engineers, 2025. pp. 1-9 (Proceedings of the International Joint Conference on Artificial Intelligence).

1.5 Thesis Structure

This thesis is organised into seven chapters, each addressing a specific aspect of the research.

Chapter 2 introduces the necessary background for the thesis. It reviews key concepts in speech processing, including automatic speech recognition (ASR), speaker verification (SV), and existing approaches to explainability in machine learning.

Chapter 3 investigates the robustness of ASR systems. It focuses on how small, human-imperceptible perturbations—motivated by auditory masking—can affect recognition outcomes, and discusses the implications for model reliability.

Chapter 4 turns to the explainability of ASR systems using post-hoc methods. It introduces a modular framework that integrates popular explanation techniques and applies them to analyze model decisions.

Chapter 5 evaluates the validity of post-hoc explanations using a phoneme recognition model with ground-truth alignment. It introduces two speech-specific variants of LIME and assesses their ability to capture meaningful attribution in the temporal structure of speech.

Chapter 6 explores intrinsic explainability in speaker verification. It proposes a concept-based framework that aligns latent dimensions with speaker attributes such as accent and age, enabling structured interpretation of verification outcomes.

Chapter 7 concludes the thesis. It summarises the main findings, discusses broader implications, and outlines potential directions for future work.

Chapter 2

Background

2.1 Introduction

Xiaoliang:new intro with details

This chapter provides the technical background required to situate and contextualise the contributions of the thesis. While Chapter 1 introduced the overall challenges, here we focus on two areas that are essential for addressing the research questions: the speech processing tasks that serve as experimental testbeds, and the explainability techniques used to interpret model behaviour. Together, these components form the technical foundation for our work on robustness and explainability in ASR and SV.

Section 2.1 reviews Automatic Speech Recognition (ASR), Phoneme Recognition (PR), and Speaker Verification (SV), which are the three tasks consistently adopted throughout the thesis. Section 2.2 introduces the main families of explainability approaches, covering representative post-hoc and intrinsic methods. Particular emphasis is placed on perturbation-based attribution, which directly motivates the investigations in Chapters 4 and 5, and on concept bottleneck models, which underpin the approach developed in Chapter 6.

Robustness against adversarial attack, although an important dimension of this thesis, is relatively confined to Chapter 3 and therefore introduced directly in that context rather than here. The present chapter is thus deliberately selective: rather than providing a comprehensive survey of the broad literatures on speech processing or explainability, it concentrates on the concepts and tools necessary for following the analyses in later chapters. In this way, it supports readers with different levels of prior knowledge while maintaining a clear line of connection to the research questions.

2.2 Speech Processing

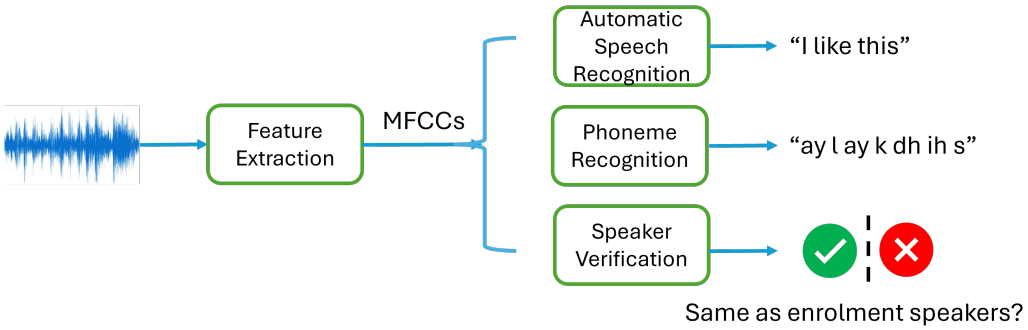


Figure 2.1: Overview of the traditional speech processing pipeline in our thesis. After feature extraction, the MFCC features can be used for multiple downstream tasks including automatic speech recognition (ASR), phoneme recognition, and speaker verification (SV). In SV, the system determines whether the input is spoken by the same person as the enrollment utterance.

In this thesis, all speech-related tasks are implemented using a traditional processing pipeline. As illustrated in Figure 2.1, this pipeline begins with standard feature extraction—specifically, Mel-Frequency Cepstral Coefficients (MFCCs)—which serve as the input to downstream models. The tasks investigated in this work include automatic speech recognition (ASR), phoneme recognition (PR), and speaker verification (SV). While these tasks are distinct in purpose, they share a common front-end.

2.2.1 MFCC Extraction

Raw speech waveforms, despite containing rich linguistic and speaker-specific information, pose several challenges when used directly in machine learning models. First, they are high-dimensional and redundant—for example, a 16 kHz recording produces 16,000 amplitude values per second, making direct processing computationally expensive and inefficient [40]. Second, raw waveforms exhibit high variability, as factors like speaker differences, background noise, and recording conditions significantly alter their shape [41]. Finally, speech is generally more informative in the frequency domain, as phonetic and speaker-related characteristics are better distinguished through spectral analysis rather than time-domain amplitude variations [42]. Due to these issues, raw waveforms are typically preprocessed into structured feature representations before further modeling.

Among the most widely used approaches are handcrafted acoustic features, which extract relevant information while reducing redundancy and variability. Mel-frequency cepstral coefficients (MFCCs) approximate human auditory perception [43], Mel filterbank energies (Melbank) retain spectral properties crucial for speech modeling [42], and linear predictive coding (LPC) captures vocal tract characteristics through autoregressive modeling [44]. These rep-

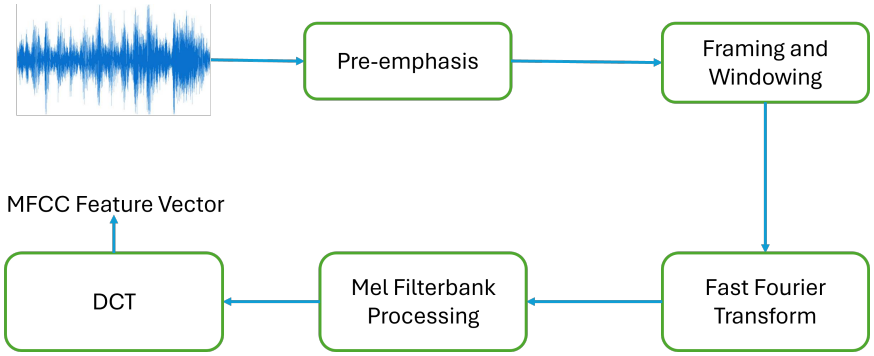


Figure 2.2: An illustration of the MFCC process.

resentations have been foundational in ASR and SV, forming the basis of traditional speech processing pipelines. While recent advances have enabled some end-to-end models to operate directly on raw waveforms [45], handcrafted features remain integral to many speech applications due to their interpretability and efficiency.

Among the various handcrafted features used in speech processing, Mel-Frequency Cepstral Coefficients (MFCCs) stand out as one of the most widely adopted and well-established representations. Given their long-standing use in ASR and SV, we choose to focus on MFCCs in this background section as a representative example of handcrafted features.

Beyond their foundational role, MFCCs are directly relevant to this thesis, as several models in our study have been trained using MFCC-based features, particularly in speaker verification and phoneme recognition. Furthermore, the MFCC extraction pipeline has served as a source of inspiration for our robustness research in Chapter 3, where we leverage similar transformations when designing adversarial attacks.

The following section provides a detailed description of the MFCC extraction process.

Feature Extraction Process

As shown in the Figure 2.2, the process of extracting MFCCs consists of the following key steps:

1. **Pre-emphasis**: Speech contains stronger low-frequency components, which can overshadow important high-frequency details. A high-pass filter is applied to emphasize higher frequencies and balance the spectrum.
2. **Framing and Windowing**: Speech is a non-stationary signal, but short segments can be treated as approximately stationary. To enable time-localized spectral analysis, the signal is divided into overlapping frames. Each frame is windowed to reduce edge effects during the frequency transformation.

3. **Fast Fourier Transform (FFT)**: Phonetic and speaker information are primarily encoded in spectral patterns. Each windowed frame is transformed from the time domain to the frequency domain using the Fast Fourier Transform (FFT), revealing the spectral content essential for further processing.
4. **Mel Filterbank Processing**: Human auditory perception is more sensitive to frequency differences in lower ranges. To mimic this, a bank of triangular filters is applied across the spectrum, spaced according to the Mel scale. These filters aggregate energy in perceptually relevant frequency bands, producing Mel-filtered spectral energies.
5. **Logarithm and Discrete Cosine Transform (DCT)**: The Mel-filtered energies are log-transformed to reflect the nonlinear perception of loudness. A Discrete Cosine Transform (DCT) is then applied to compact the information and produce decorrelated coefficients.

These steps together produce the final Mel-Frequency Cepstral Coefficients (MFCCs), which are widely used as robust and interpretable features in speech processing systems.

2.2.2 Automatic Speech Recognition (ASR)

ASR is the computational process of converting spoken utterances into corresponding sequences of words. In Chapters 4 and 5 of this thesis, we adopt traditional and well-established ASR architectures rather than modern end-to-end systems. This choice is motivated by the current state of explainability research in speech processing: the field remains relatively underexplored, and the lack of established benchmarks makes it difficult to evaluate explanation methods in a reproducible manner. By using classical components in a modular setup, we construct a reliable environment that serves as an initial testbed—one that lays the groundwork for future exploration with more complex, state-of-the-art systems.

The classical modular architecture consisting of three primary components: an acoustic model, a pronunciation lexicon, and a language model [42].

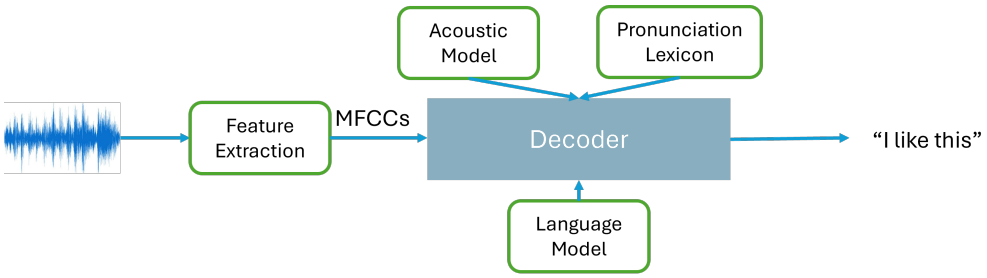


Figure 2.3: An illustration of the modular architecture of ASR.

As shown in Figure 2.3:

1. **Acoustic Model**
   The acoustic model is responsible for analyzing short segments of audio and estimating the likelihood of different hidden Markov model (HMM) states at each moment in time. These HMM states are typically tied to sub-phonetic units, such as parts of phonemes(i.e., the smallest unit of sound that can distinguish meaning in a language), and represent the basic building blocks of speech sounds. Instead of directly predicting full phonemes or words, the acoustic model tells us how likely it is that the audio corresponds to each possible HMM state at each time step. In earlier systems, this estimation was done using statistical models like GMM-HMMs [46]. Modern systems [47] use deep neural networks to produce more accurate estimates of these state likelihoods based on input features.
2. **Pronunciation Lexicon**: The lexicon defines how each word in the vocabulary is pronounced in terms of phoneme sequences, and indirectly in terms of HMM states. For example, the word “like” may be represented as a sequence of phonemes /l/, /ay/, /k/, with each phoneme further broken into a small set of states. This mapping allows the system to link low-level acoustic evidence to candidate words.
3. **Language Model**: Because multiple word sequences might share similar or identical phoneme sequences, the language model provides essential context. It scores how likely different sequences of words are in the target language. For instance, if the acoustic evidence supports either “I like this” or “Eye like this,” the language model helps prefer “I like this” because it is more common and grammatically appropriate.

During decoding, the system integrates these components: it takes the frame-by-frame HMM state likelihoods from the acoustic model, maps possible state sequences to word sequences using the lexicon, and then evaluates the final hypotheses using the language model. The decoder integrates these scores and selects the word sequence with the highest overall likelihood.

2.2.3 Phoneme Recognition as a simplified ASR Task

ASR systems are designed to transcribe spoken language into word sequences. Phoneme recognition (PR), in contrast, targets the prediction of phoneme sequences—where a phoneme is the smallest contrastive unit of sound in a language, capable of distinguishing meaning between words (e.g., /b/ vs. /p/ in bat vs. pat) [48]. Although the output differs, the overall structure of the system remains the same, with an acoustic model followed by decoding. The key distinction lies in the simplification of the decoding process. Rather than mapping acoustic inputs to words from a large vocabulary, PR systems operate on a fixed set of phonemes and typically use a minimal lexicon and a simpler, lower-order language model. These adjustments reduce the influence of higher-level linguistic constraints, allowing the acoustic model to play a more prominent role in determining the final output.

This makes phoneme recognition particularly useful for controlled analysis. It isolates the acoustic modeling component while preserving the overall pipeline structure, enabling clearer observation of model behavior.

To evaluate the validity of post-hoc explanation methods, we include a phoneme recognition task as part of our study in Chapter 5. Compared to full ASR systems, phoneme recognition offers a simplified and more controlled setting by isolating the acoustic model from higher-level components such as the pronunciation lexicon and language model.

2.2.4 Speaker Verification (SV)

Speaker verification (SV) is the task of determining whether two speech recordings originate from the same speaker. Unlike speaker identification, which assigns an input utterance to one of many known speaker identities, speaker verification performs a binary classification: it verifies whether a test utterance matches the claimed identity based on a comparison with an enrollment utterance. The system outputs a similarity score between the two utterances, and if the score exceeds a predefined threshold, the identity claim is accepted; otherwise, it is rejected [49, 50].

SV systems are widely deployed in biometric authentication, forensic analysis, and personalized human-computer interaction [51, 52]. Early approaches such as Gaussian Mixture Models with Universal Background Models (GMM-UBM) [49] and i-vector frameworks [51] represented speaker characteristics using low-dimensional, statistical embeddings derived from sufficient statistics of speech segments. These methods typically relied on generative modeling and session compensation techniques. Recent advances in deep learning have led to the dominance of neural speaker embeddings, where a neural network is trained to extract fixed-dimensional vectors—known as speaker embeddings—that capture speaker-specific characteristics while discarding irrelevant factors such as phonetic content or noise.

Those embeddings are then compared using similarity metrics such as cosine similarity or Probabilistic Linear Discriminant Analysis (PLDA) [53, 54].

In the Chapter 6, we investigate how group-level speaker attributes influence the decisions made by speaker verification models. Rather than focusing solely on overall performance metrics, we aim to understand whether specific speaker characteristics—such as gender, age, or accent—systematically affect verification outcomes. This perspective shifts the focus from aggregate model performance to the internal decision behavior of SV systems, contributing to ongoing discussions around fairness and explainability. To support this analysis, we utilize two classic and widely-used deep speaker embedding architectures: x-vector [53] and ECAPA-TDNN [54]. A brief overview of both models is provided below.

Overview of X-vector and ECAPA-TDNN Architectures

Both x-vector and ECAPA-TDNN are widely used architectures for extracting fixed-dimensional speaker embeddings from variable-length speech utterances. They share a high-level structure that includes three stages: frame-level feature encoding, utterance-level pooling, and speaker classification during training. However, ECAPA-TDNN incorporates several enhancements that improve speaker discriminability and robustness over the original x-vector framework.

The x-vector model uses a time-delay neural network (TDNN) [53] to capture short-term temporal patterns from frame-level acoustic features. These frame-level representations are then summarized using a simple statistical pooling layer—typically computing the mean and variance over time—to produce a fixed-length utterance-level vector. This vector passes through several fully connected layers, and the final speaker embedding is usually extracted from the last hidden layer before the softmax classification layer. The model is trained with a standard softmax loss to classify speaker identities.

ECAPA-TDNN builds on this architecture with several key improvements. It replaces plain TDNN layers with Res2Net blocks [55], enabling multi-scale temporal feature extraction within each layer. It also introduces squeeze-and-excitation (SE) modules [56] to recalibrate channel-wise feature importance, helping the model focus on speaker-relevant information. Instead of relying on a single frame-level layer for pooling, ECAPA aggregates features from multiple layers—a process known as multi-layer feature aggregation. Additionally, it adopts an attentive statistical pooling mechanism that assigns different importance weights to different frames. The model is typically trained with an angular margin loss, such as additive margin softmax (AM-Softmax) [57], to enforce tighter intra-class clustering and better inter-class separation. For implementation details, we refer readers to the original papers [53, 54].

2.3 Explainability in Artificial intelligence

Having introduced the key components of speech processing relevant to this thesis, we now turn to the second major area: explainability. This section outlines the foundational concepts and main categories of explainability methods. Rather than covering the full landscape of explainability research, we highlight only the aspects necessary to understand and the methods used in this thesis.

AI has made substantial progress in recent years, particularly in areas such as natural language processing (NLP) and computer vision. Among various approaches, deep learning has emerged as a dominant paradigm, delivering state-of-the-art performance across a wide range of tasks. Despite these advances, the opacity of deep learning models has become a growing concern. Unlike models such as linear classifiers or decision trees–where the

reasoning behind predictions can be directly traced through weights or explicit decision paths–deep learning models often function as black boxes, offering little insight into how they arrive at their outputs. This lack of transparency raises important issues related to trust, fairness, accountability, and regulatory compliance. As a result, there is an increasing demand for AI systems that are not only accurate but also capable of explaining their decisions in a manner that is understandable to humans.

Despite its importance, the term "explainability" lacks a unified definition. Some researchers emphasize transparency—whether a model's internal mechanisms are inherently interpretable—while others focus on human-comprehensibility, asking whether explanations improve a user's understanding and trust in the model's decisions.

Lipton [58] categorizes explainability into three key dimensions:

- **Simulatability**: Whether a human can mentally simulate the model's reasoning entirely.
- **Decomposability**: Whether each part of the model (features, parameters, or components) is interpretable.
- **Algorithmic Transparency**: Whether the underlying algorithm is mathematically understandable.

Alternatively, Doshi-Velez and Kim [59] propose a user-centric view, defining explainability as "the degree to which a human can understand and predict a model's decisions." This emphasizes the practical utility of explanations in supporting human decision-making. These perspectives lead to the commonly accepted distinction between **Post-hoc Explainability** and **Intrinsic Explainability**:

- **Post-hoc Explainability**: Methods that explain the decisions of complex, black-box models after predictions have been made.
- **Intrinsic Explainability**: Models that are interpretable by design (e.g., decision trees, linear models).

Some works distinguish between "explainability" and "interpretability," using the former to refer to post-hoc explanations and the latter to describe models that are inherently transparent. However, this distinction is not consistently defined across the literature. So in this thesis, we do not make a strict separation between the two terms and use them interchangeably.

To systematically illustrate how different explainability methods interpret model decisions, we introduce a representative example that will be revisited throughout this chapter:

Example: Consider the sentence

"The salary is low, and the workload is exhausting."

which a sentiment analysis model predicts as expressing negative sentiment. We will demonstrate how various methods analyze this decision.

2.3.1 Post-hoc Explainability

Post-hoc explainability methods are designed to provide insights into the decision-making processes of black-box models after a prediction has been made.

Over the past decade, researchers have proposed a wide range of post-hoc methods to interpret such models. Among these, feature attribution has emerged as one of the earliest developed and most widely adopted categories. By assigning importance scores to input features, feature attribution methods provide intuitive and actionable explanations that have been successfully applied across various domains, including computer vision [60, 61, 62, 34, 63, 36] and natural language processing [64].

In this thesis, we begin our exploration of post-hoc explainability by applying representative feature attribution methods to ASR systems, as detailed in Chapter 4. The specific methods adopted, along with the reasons for their selection, are discussed in the following sections.

Feature Attribution Methods

Feature attribution methods explain model predictions by assigning an importance score to each feature in the input, indicating its contribution to the output. For text inputs, this might involve assigning a relevance score to each word. Using the example introduced earlier, in the sentence “The salary is low, and the workload is exhausting.”, the explanation will consist of importance scores assigned to each word based on its relevant influence on the negative sentiment prediction. Words like “low” and “exhausting” might be expected to receive higher scores, while neutral words such as “salary” and “workload” should have lower contributions.

It is important to note that the definition of a “feature” in attribution methods is not fixed. In text, a feature may correspond to a word, a subword, or a span of words; in images, it may refer to a pixel or a patch. In this discussion, we use word-level features to simplify the presentation and make the explanation more accessible to the reader.

Two main types of feature attribution methods are widely used: gradient-based and perturbation-based.

Gradient-Based Attribution. Gradient-based methods explain model predictions by computing the gradient of the model’s output with respect to each input feature, using the gradient magnitude as an indicator of feature importance. This approach relies on the assumption that the input space is continuous and scale-invariant, such that small changes in the input lead to meaningful changes in the output. This assumption is naturally satisfied in image tasks, where inputs consist of continuous pixel values. In text models, however, inputs are discrete tokens. Although gradient-based attribution has been applied to text by computing

gradients with respect to word embeddings [60], the structure of the embedding space does not always reflect semantic continuity. In this thesis, we continue to use text-based examples for consistency and clarity, while acknowledging that the theoretical assumptions underlying gradient-based attribution are more naturally aligned with image inputs.

The most basic method is the Saliency Map [65], which directly visualizes these gradients. However, due to the complex behavior of deep neural networks, such gradients are often noisy or too small to reveal meaningful patterns, making the explanations difficult to interpret.

To improve stability and theoretical grounding, Integrated Gradients [60] estimate importance by averaging gradients along a path from a baseline input to the actual input. This method satisfies desirable properties such as sensitivity and implementation invariance. SmoothGrad [66] enhances visual clarity by averaging saliency maps from multiple noisy versions of the input.

Grad-CAM [61] is another widely used technique, particularly for convolutional networks. It generates class-specific saliency maps by combining feature maps from intermediate layers with gradient information from the target class. This highlights spatial regions most relevant to the predicted label and has been widely applied to both image and audio models.

Despite their popularity, gradient-based attribution methods have important limitations. They rely on access to the model's internal structure to compute gradients of the output with respect to the input via backpropagation. Rather than directly modifying the input, these methods estimate feature importance based on how small input changes would affect the output, as inferred from the model's gradients. This makes them unsuitable for non-differentiable models or deployment settings, such as commercial ASR systems, where internal computation is inaccessible.

In this thesis, we did not apply gradient-based methods to ASR models because many production-level ASR systems expose only their inference interfaces, without providing access to model internals or gradient information, making such methods infeasible in practical scenarios.

Perturbation-Based Attribution. In contrast, perturbation-based attribution methods assess feature importance by directly modifying the input and observing how the output changes. These methods do not require access to the model's internal structure or gradients, making them applicable to black-box models such as commercial ASR APIs. These methods operate by perturbing input features and measuring the effect on the prediction.[1] The underlying assumption is that important features will significantly influence the prediction when altered or removed, while less relevant features will have little impact.

1. The definition of a feature depends on the data modality and the level of granularity adopted in the analysis. For instance, in images, a feature may correspond to a single pixel or a patch; in text, to a subword unit, a word, or a phrase.

Using the previously introduced example sentence "`The salary is low, and the workload is exhausting.`", which a sentiment analysis model predicts as expressing negative sentiment, we can illustrate the general idea behind perturbation-based methods. These methods analyze the decision by removing or masking individual words and observing how the model's confidence in the negative sentiment changes. For instance, if removing the word "`exhausting`" substantially reduces the predicted probability of negative sentiment, this suggests that "`exhausting`" is a highly influential feature. Conversely, if removing "`salary`" has little effect, its contribution is likely minimal.

It should be noted that while this example demonstrates the core intuition, different methods adopt distinct strategies for quantifying feature importance based on these perturbations. In the following, we briefly review several representative perturbation-based attribution techniques. A subset of these methods is further applied in Chapter 4 to analyze ASR models in a black-box setting.

Local Interpretable Model-agnostic Explanations (LIME)

One of the earliest and most influential perturbation-based methods is Local Interpretable Model-agnostic Explanations (LIME)[34]. LIME was proposed to address the difficulty of interpreting complex black-box models by approximating their local behavior with a simpler, interpretable model. The core assumption is that while the global decision boundary may be highly nonlinear, the model can often be well-approximated by a linear function in the neighborhood of a specific input. Based on this idea, LIME explains individual predictions by generating perturbed versions of the input and fitting a sparse linear regressor to mimic the model's output locally. In text classification, this involves randomly removing words and recording the impact on the prediction. LIME produces intuitive, model-agnostic explanations and has been widely adopted across various domains. For these reasons, we selected LIME as a baseline method for our explainability experiments on ASR systems in Chapter4.

Shapley-Value-Based Attribution. Following the development of LIME, which provides intuitive but relatively simple explanations, SHAP (SHapley Additive exPlanations) [67] introduced a more theoretically grounded approach based on cooperative game theory, which studies how to fairly distribute a collective outcome among multiple contributors. SHAP attributes the model's prediction to each feature by calculating its average marginal contribution across all possible subsets of features, offering a principled measure of feature importance.

Although SHAP has become a widely recognized and influential method, it suffers from significant practical limitations. Computing exact Shapley values is computationally intractable for high-dimensional inputs.

Given these computational challenges and the characteristics of ASR tasks, we did not adopt SHAP in our experiments. ASR inputs typically consist of long sequences of acoustic features, leading to a combinatorially large number of feature subsets. This is in sharp contrast to applications like text sentiment analysis, where the number of words in a sentence is usually small, and the computational cost remains manageable. In ASR scenarios, the exponentially growing subset space makes SHAP prohibitively time-consuming, rendering it impractical for real-world usage.

Statistical Fault Localization

Sun et al. proposed a fundamentally different approach by introducing Statistical Fault Localization (SFL) [63] for model interpretability in their work [35]. Borrowed from software engineering, where SFL techniques identify faulty program components responsible for system failures, this approach analyzes statistical patterns between feature perturbations and output changes to determine feature importance.

Unlike LIME and SHAP, which rely on surrogate models or exhaustive subset evaluation, SFL directly measures the statistical association between input feature modifications and the model's output behavior. Classic suspiciousness metrics from fault localization—such as Tarantula [68]—are employed to rank feature importance. These metrics quantify how frequently the presence or absence of a feature correlates with significant changes in the model's output when that feature is perturbed.

Experimental results in the original paper demonstrated that SFL achieved more faithful and stable explanations than both LIME and SHAP. At the time of our experiments, SFL remained one of the most effective and efficient methods available. For these reasons, we selected SFL as a key method in our explainability analysis for ASR systems.

Causal Reasoning and Counterfactual Explanations.

As perturbation-based attribution methods continued to evolve, researchers increasingly recognized the limitations of purely correlational analyses and began shifting toward causal reasoning frameworks. Instead of merely identifying which features contribute to a model's current prediction, causal attribution methods aim to answer a more insightful question: what minimal changes to the input would causally lead to a different outcome? For example, still consider the sentence `"The salary is low, and the workload is exhausting."` Correlational methods may highlight both `"low"` and `"exhausting"` as important, but they do not reveal whether the prediction would actually change if one of these words were removed. A causal method, by contrast, might determine that removing `"exhausting"` is sufficient to flip the prediction, while removing `"low"` has little effect—thereby providing a clearer view of how the model's decision boundary is structured in this local region.

Early counterfactual explanation methods, such as the work by Wachter et al. [69], formulated the problem as an optimization task, identifying the smallest input modifications needed to change a model's decision. However, these approaches often resulted in implausible or semantically invalid counterfactuals—an issue particularly severe for sequential data like speech, where low-level perturbations rarely produce meaningful or realistic variations.

To improve the plausibility of generated counterfactuals, later methods introduced additional constraints. For example, DiCE [70] proposed generating diverse and actionable counterfactuals, offering users multiple realistic alternatives.

Building on these developments, **Causal**, proposed by Chockler et al. [36], introduced a principled framework specifically designed for causal explanation in black-box models. This work adopts the actual causality theory of Halpern and Pearl [71] and establishes a complete formal framework for defining and discovering causal explanations. It focuses on identifying minimal sets of input features that constitute sufficient causes for a model's decision.

Due to its well-established theoretical foundation, clear formal definitions, and complete algorithmic design, this framework offers a practical and theoretically sound solution for discovering causal explanations. These characteristics make it particularly suitable for our purposes, and thus, it was directly adopted in our chapter 4.

2.3.2 Post-hoc Perturbation-Based Explainability Methods: A Unified Perspective

Perturbation-based methods represent one of the most widely used families of explainability techniques. While later chapters adapt and extend some of these methods as part of the thesis contributions, it is useful here to introduce a unified conceptual perspective as background. The aim is not to propose a new algorithm, but to synthesise common principles that underlie approaches such as LIME, SFL, and causal attribution. By presenting them within a single perspective at this stage, readers are provided with a clearer foundation for understanding how these techniques will be compared and applied in subsequent chapters.

The preceding discussion has focused on the theoretical foundations of post-hoc explainability methods and provided the rationale for selecting three representative approaches: LIME [34], SFL [63] and Causal [36]—for application on ASR. We now shift to the technical foundation used in this thesis. In particular, we conclude a unified framework for these three perturbation-based methods. This framework uses formal notation to systematically describe the core steps shared by these methods, enabling a clearer comparison of their underlying mechanisms and supports future extensions.

Perturbation-based methods share a common framework centered on modifying the input and observing how these changes affect the model's predictions. The general workflow includes four key steps:

- **Generate Perturbations:** Modify features[2] of the input to create variations.
- **Classify Mutants:** Test the model on these modified inputs and record how the predictions change.
- **Quantify Importance:** Analyze how each modified part impacts the output to determine its importance.
- **Construct Explanation:** Use the importance scores to rank input parts or find the smallest set of input parts needed to generate the original output.

Among the selected three techniques, their primary differences lie in how they handle the third step of quantifying importance. Below, we briefly describe their key characteristics in this step.

- LIME quantifies importance by approximating the model's local behavior using a simple, interpretable model, such as a linear model. The parameters of this surrogate model are treated as importance scores.
- SFL is the first to apply statistical fault localization techniques to compute importance. Unlike methods that fit a surrogate model to approximate local behavior, SFL directly estimates feature importance using statistical associations between feature presence and model prediction outcomes.
- Causal methods are grounded in causality theory [71], leveraging principles of cause-and-effect relationships to identify input components responsible for changes in the model's output.

Unified Notation

- *Input Representation:* Let the input be represented as $\mathbf{x}$, where the structure of $\mathbf{x}$ depends on the data modality like text and image:
  - *Text:*
    $$\mathbf{x} = [x_1, x_2, \ldots, x_n], \quad x_j \in \mathscr{X}^*, \tag{2.1}$$
    where x_j is the j-th word or token, and n is the total number of tokens in the sequence. The input x_j are assumed to be indexed sequentially, such that $x_1, x_2, \ldots, x_n$ are adjacent but non-overlapping.
  - *Images:*
    $$\mathbf{x} = [x_{i,j}] \in \mathbb{R}^{h \times w},$$
    where h and w are the height and width of the image, respectively, and $\mathbf{x}_{i,j}$ represents the pixel intensity at row i and column j.
    For simplicity, throughout this section, we use the text-based representation of $\mathbf{x}$, where $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ and x_j represents the j-th input element.

2. The definition of a feature depends on the data modality and the level of granularity adopted in the analysis. For instance, in images, a feature may correspond to a single pixel or a patch; in text, to a subword unit, a word, or a phrase.

- *Output Representation:* The black-box model f maps the input $\mathbf{x}$ to a scalar output y:

$$f : \mathscr{X}^* \to \mathscr{Y},$$

where:

  - $\mathscr{X}^*$ is the input space, which may include sequences of varying lengths.
  - $\mathscr{Y}$ is the set of possible labels (e.g., classification categories).
  - $y = f(\mathbf{x})$ is the predicted label for the input $\mathbf{x}$.

Example: Consider a black-box model f trained for sentiment classification. The model takes a text input $x = [x_1, x_2, \ldots, x_n]$, where each x_j is a word or token, and outputs a label $y \in \mathscr{Y}$. The set of possible labels is:

$$\mathscr{Y} = \{\texttt{positive}, \texttt{neutral}, \texttt{negative}\}.$$

For illustration purposes within this framework, we select a relatively short example sentence to keep the explanation readable.

Specifically, consider the input $x =$ [This, exam, is, difficult]:

- $n = 4$, and each x_j represents a word (e.g., $x_1 =$ This, $x_4 =$ difficult).
- The model prediction $y = f(x)$ could be $y = \texttt{negative}$, indicating a negative sentiment.

Step 1: Generate Perturbations (Mutants)

Perturbations, or mutants, are generated by masking parts of the input $\mathbf{x}$. Here, masking refers to modifying selected parts of the input and replacing them with a predefined value (e.g., a "*" token for a word in text[3]). The specific masking strategy varies depending on the method:

- *LIME and SFL:* A subset of input elements x_j is masked randomly, determined by a masking ratio $\alpha \in (0,1)$. The size of the masked subset is:

$$|S| = \lceil \alpha n \rceil, \quad S \subseteq \{1, 2, \ldots, n\}, \tag{2.2}$$

where S represents the indices of the elements to be masked. The perturbed input $\mathbf{x}' = [x'_1, x'_2, \ldots, x'_n]$ is then computed as:

$$x'_j = \begin{cases} \bar{x}_j, & \text{if } j \in S, \\ x_j, & \text{otherwise,} \end{cases} \qquad \forall j \in \{1, 2, \ldots, n\}, \tag{2.3}$$

3. In practice, masking can be implemented by replacing the word's embedding with a zero vector. The "*" symbol is used here for readability to represent such a masked position.

where $\bar{x}_j$ is a predefined value used to replace the masked elements, such as a "*" token for text, or a uniform background color for images. The indices of the masked elements in S are selected randomly, and the elements do not need to be contiguous. Example: Consider x = [This, exam, is, difficult, today], with $n = 5$. Suppose the masking ratio is $\alpha = 0.4$, so:

$$|S| = \lceil 0.4 \times 5 \rceil = 2.$$

If $S = \{1, 5\}$, the masked elements are "exam" and "today," resulting in a mutant:

$$x' = [*, \text{exam} , \text{is}, \text{difficult}, *].$$

However, in many cases, adjacent elements contribute more effectively as a group than individually. For example, in text, phrases such as "very difficult" often carry a stronger semantic meaning than the individual words "very" and "difficult" alone. To address this, Causal considers contiguous groups of elements when generating perturbations.

- *Causal:* To account for interactions between neighboring elements, the input $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ is divided into m non-overlapping and contiguous regions $\{C_1, C_2, \ldots, C_m\}$ $(1 \leq m \leq n)$, where each region C_i consists of consecutive elements and is defined as:

$$C_i = \{x_{k_i}, x_{k_i+1}, \ldots, x_{k_i+|C_i|-1}\}, \tag{2.4}$$

where $|C_i|$ is the number of elements in C_i, and k_i is the starting index of region C_i. The starting indices satisfy:

$$k_1 = 1, \quad k_{i+1} = k_i + |C_i|, \quad \forall i \in \{1, \ldots, m-1\}. \tag{2.5}$$

The regions satisfy the following conditions:

$$\bigcup_{i=1}^{m} C_i = \{x_1, x_2, \ldots, x_n\}, \quad C_i \cap C_j = \emptyset, \forall i \neq j. \tag{2.6}$$

Perturbation involves masking a proportion $\alpha \in (0, 1)$ of the regions $\{C_1, C_2, \ldots, C_m\}$. Similar to the process in SFL and LIME, the mask is applied at the level of regions rather than individual elements. Specifically, if a subset of regions is selected, all elements within these regions will be masked, while the remaining regions are left unchanged.

Step 2: Classify Mutants

Each mutant $\mathbf{x}'$ is evaluated by the black-box model f, producing a prediction $y' = f(\mathbf{x}')$, where $y = f(\mathbf{x})$ is the original prediction. The mutant is classified as:

$$\begin{cases} \text{Success}, & \text{if } y' = y, \\ \text{Failure}, & \text{if } y' \neq y. \end{cases} \tag{2.7}$$

where $y = f(\mathbf{x})$ is the original prediction.

Step 3: Quantify importance

This step is where three perturbation-based techniques differ most, since each method uses distinct principles to evaluate how input changes affect the model output.

- **LIME** quantifies the importance of input components by treating each element x_j of the input $\mathbf{x}$ as a feature and fitting a simple linear model to approximate how much its presence or absence affects the model's prediction. The resulting weights from the linear model directly serve as importance scores, indicating each element's contribution.
  The formal process is described below.
  Each perturbed input(mutant) $\mathbf{x}'$ is represented as a binary vector $\mathbf{z} = [z_1, z_2, \ldots, z_n]$ of the same length as the original input $\mathbf{x}'$, where:

$$z_j = \begin{cases} 1, & \text{if the } j\text{-th component is unmasked}, \\ 0, & \text{if the } j\text{-th component is masked}. \end{cases} \tag{2.8}$$

  To focus on mutants that are similar to the original input, LIME assigns a weight $w(\mathbf{z})$ to each mutant:

$$w(\mathbf{z}) = \exp\left(-\frac{\text{dist}(\mathbf{z}, \mathbf{1})^2}{\sigma^2}\right), \tag{2.9}$$

  where $\text{dist}(\mathbf{z}, \mathbf{1}) = \sum_{j=1}^{n}(1 - z_j)$ measures the number of masked components in $\mathbf{z}$, and σ controls how much emphasis is placed on mutants close to the original input. Mutants that are closer to the original input (dist is small) are given higher weights because the changes they introduce are more localized and easier to attribute to specific components. For example, observing the effect of masking a single input component provides more precise information about its importance compared to masking many components at once, which can obscure individual contributions. This weighting ensures that the surrogate model focuses on capturing the local behavior of $f(\mathbf{x})$.
  The surrogate model $g(\mathbf{z})$ is then trained to fit the predictions of the black-box model $f(\mathbf{x}')$, where $\mathbf{x}'$ is the perturbed input corresponding to $\mathbf{z}$. The training process minimizes a weighted loss function:

$$\mathscr{L}(f,g,w) = \sum_{\mathbf{z}\in\mathscr{Z}} w(\mathbf{z}) \cdot (f(\mathbf{z}) - g(\mathbf{z}))^2, \tag{2.10}$$

where $\mathscr{Z}$ refers to the full set of mutants selected and generated, and $w(\mathbf{z})$ ensures that mutants closer to the original input receive higher importance during fitting.
The surrogate model is expressed as:

$$g(\mathbf{z}) = \beta_0 + \sum_{j=1}^{n} \beta_j z_j, \tag{2.11}$$

where β_j represents the importance score for the j-th input component. These scores quantify how much each component contributes to the model's prediction. Larger $|\beta_j|$ values correspond to stronger influences on the output, providing a clear explanation of the input-output relationship.

- **SFL** quantifies the importance of each input element x_j by analyzing how often masking it causes the model's prediction to change. Elements that lead to more prediction failures when masked are considered more important.
  The importance score for feature x_j, denoted by ϕ_j, is defined as:

$$\phi_j \propto \frac{n_{\mathsf{f}}^j}{n_{\mathsf{f}}^j + n_{\mathsf{p}}^j}, \tag{2.12}$$

  To calculate $s(x_j)$, SFL collects the following statistics for each x_j:
  – n_{f}^j: Number of mutants where x_j is masked and the prediction changes.
  – n_{p}^j: Number of mutants where x_j is masked and the prediction remains unchanged.
  Higher scores indicate that masking x_j has a stronger influence on the output.
  The equation above serves as a base for many statistical measures used in SFL, with each measure incorporating specific adjustments. For detailed explanations and examples of these measures, we recommend referring to the paper [63].
- **Causal** methods evaluate the importance of each input region based on its ability to causally influence the model's output. Specifically, these methods measure whether masking a region C_i alone can change the model's prediction. If so, the region is considered highly important. If the model's output remains unchanged, the method assesses how many additional regions must be masked together with C_i to alter the prediction.
  Let $\mathscr{C} = \{C_1, \ldots, C_{i-1}, C_{i+1}, \ldots, C_m\}$ denote the set of all input regions excluding C_i. Let $O \subseteq \mathscr{C}$ be any subset of additional regions. We define r_i as the minimum number of additional regions that must be masked alongside C_i to change the model's output:

$$r_i = \min\{|O| : O \subseteq \mathscr{C},\ f(\mathsf{mask}(C_i \cup O, \mathbf{x})) \neq f(\mathbf{x})\}, \tag{2.13}$$

where $f(\cdot)$ is the model's prediction function and mask$(C_i \cup O, \mathbf{x})$ denotes the input after masking the regions C_i and O.

The importance of region C_i is then quantified by the *Responsibility* score ψ_i, defined as:

$$\psi_i = \frac{1}{r_i + 1}. \tag{2.14}$$

This metric reflects a region's direct influence on the prediction: if masking C_i alone changes the output ($r_i = 0$), then $\psi_i = 1$, indicating maximum importance. If more regions need to be masked ($r_i > 0$), then $\psi_i < 1$, decreasing as more support is needed to cause an effect.

Step 4: Construct Explanation

This step aims to construct an explanation set $\mathscr{E}$ that contains the most important elements or regions needed to reproduce the model's prediction $f(\mathbf{x})$.

For **LIME**, the method outputs an importance score β_j for each input feature x_j, reflecting how much it contributes to the prediction. However, it does not directly provide a minimal set of elements sufficient to explain the outcome.

In contrast, both **SFL** and **Causal** explicitly aim to build a compact explanation set $\mathscr{E}$ by selecting the smallest subset of input features that still preserves the model's original prediction.

In **SFL**, each input feature x_j is assigned a statistical importance score ϕ_j. The elements are ranked by ϕ_j, and added one by one to $\mathscr{E}$ in descending order until the model's prediction based on $\mathscr{E}$ matches the original output.

In **Causal**, the input is divided into regions $\{C_1, C_2, \ldots, C_m\}$, and each region C_i is given an importance score ψ_i. Regions with higher ψ_i are added first, as they have stronger causal influence on the output.

Formally, the goal is to find the smallest $\mathscr{E} \subseteq \{x_1, \ldots, x_n\}$ (or subset of regions, depending on the method) such that:

$$f(\text{mask}(\overline{\mathscr{E}}, \mathbf{x})) = f(\mathbf{x}), \tag{2.15}$$

where $\overline{\mathscr{E}}$ denotes the set of components not in $\mathscr{E}$, and mask$(\overline{\mathscr{E}}, \mathbf{x})$ is the input with non-$\mathscr{E}$ parts masked.

This formulation ensures that only the most relevant parts of the input are retained, providing a faithful and compact explanation of the model's decision.

2.3.3 Intrinsic Explainability

In contrast to post-hoc methods, which generate explanations after a model makes a prediction, intrinsic explainability focuses on designing models whose internal reasoning is inherently transparent. These models structure their decision processes in a way that directly incorporates human-understandable process, allowing their outputs to be interpreted without relying on external explanation techniques [58, 59].

To maintain consistency with earlier discussions, we continue using the example sentence:

"The salary is low, and the workload is exhausting."

While post-hoc methods identify influential words after a decision is made, intrinsic methods integrate explainability directly into the prediction process. Depending on the approach, this can be achieved through explicit rules, reference examples, or intermediate human-interpretable concepts.

Rule-Based Models Rule-based models predict outcomes by applying explicit logical rules. For the example sentence, a rule-based sentiment classifier might encode logic such as: *If "low" appears near "salary" and "exhausting" appears near "workload", predict negative sentiment.*

Such models offer complete transparency, as every decision path is traceable [72]. However, rule-based systems become impractical for complex tasks like Speaker Verification, where decision boundaries are formed over continuous, high-dimensional inputs, and symbolic rules cannot effectively capture nuanced speaker traits.

Prototype-Based Models Prototype-based models explain decisions by comparing inputs to representative examples from the training data [73]. In sentiment analysis, for instance, the model might justify a negative sentiment prediction by retrieving a similar prototype sentence such as: *"The benefits are poor, and the hours are long."*

This provides intuitive, example-based explanations, especially effective in vision and text domains [74, 75]. However, in speaker verification, speaker identity is encoded through subtle variations across high-dimensional embeddings, making it difficult to define semantically meaningful and universally applicable prototypes.

Sparse Models and Concept Bottleneck Models Sparse models achieve interpretability by limiting the number of features or concepts involved in a prediction. Among these, the Concept Bottleneck Model (CBM) [76] has attracted significant attention due to its structured decision process. CBMs decompose the prediction pipeline into two stages: first, the model predicts a set of human-defined concepts; then, it bases the final decision solely on these interpretable concepts.

To illustrate how a CBM works, consider again the sentiment classification example: *"The salary is low, and the workload is exhausting."*. Instead of predicting the sentiment label directly from the input sentence, a CBM first maps the input to a set of predefined, human-interpretable concepts. For this example, the model might predict whether the sentence expresses *financial dissatisfaction*, *physical exhaustion*, or *social isolation*. Each of these concepts is inferred individually from the input, resulting in an explicit concept vector that represents the model's intermediate understanding.

In the second stage, the model uses this concept vector to determine the final sentiment label. For instance, if both *financial dissatisfaction* and *physical exhaustion* are detected, the model may classify the sentiment as *negative*. This two-stage structure decomposes the reasoning process into concept recognition and label inference, allowing each step to be independently examined.

By separating prediction into interpretable stages, CBMs offer full transparency into the model's decision-making process. Users can observe which concepts were identified and understand how these concepts contributed to the final prediction. This transparency improves explainability, facilitates error analysis, and helps build user trust in the model's behavior [77].

CBMs are particularly suitable for scenarios where decisions naturally align with human-understandable attributes. In this thesis, we adopt CBMs to explore intrinsic explainability in Speaker Verification, where speaker characteristics—such as accent, nationality, or age—serve as meaningful intermediate concepts. This makes it possible to express verification outcomes in terms of explicit, human-relevant attributes, providing a more transparent and accountable decision process [78].

2.4 Conclusion

This thesis investigates two fundamental challenges in speech-based AI systems: robustness and explainability. To support this investigation, this chapter first introduced the speech processing tasks that serve as experimental testbeds—ASR, phoneme recognition, and SV—all implemented using traditional pipelines to ensure a controlled setup. We then reviewed key approaches to explainability, covering both post-hoc and intrinsic methods, and concluded

with a unified perturbation-based framework to structure our analysis. As for robustness, since it constitutes only a single chapter in this thesis, we integrate the necessary background directly into the next chapter alongside the corresponding project. These foundations enable the systematic exploration of robustness and explainability in the following chapters.

Chapter 3

Robustness: Blackbox Adversarial Attacks on ASR using Frequency Masking

This chapter focuses on the robustness of ASR systems through the lens of black-box adversarial attacks. Rather than aiming to improve model resilience directly, we propose a method that exposes vulnerabilities by applying imperceptible perturbations to the input audio. The chapter is organized as follows: we begin with an introduction, followed by a review of related work, then describe the proposed methodology and experimental setup, and finally present and discuss the results.

3.1 Introduction

This section provides an introduction to the current chapter and should be distinguished from the general introduction of the thesis.

The computational core of ASR are deep neural networks (DNNs) that have been shown to be susceptible to adversarial perturbations; easily misused by attackers to generate malicious outputs [79, 80, 22].

Existing work on ASR adversarial attacks. Adversarial perturbations[1] were first presented by Szegedy et al. to demonstrate the lack of robustness in DNN models – a small perturbation of an input may lead to a significant perturbation of the output of a DNN model [81]. This vulnerability can be exploited by adversaries to augment the original input with a crafted perturbation, invisible to a human but sufficient for the DNN model to misclassify this input. This influential work triggered several research contributions in the computer vision domain that generate adversarial attacks for testing security and robustness of vision tasks [82, 83, 84]. Research on the use of adversarial attacks on ASR is, however, only just emerging, and can

1. Also referred to as Adversarial examples or Adversarial attacks.

be classified along two dimensions,

1. Un-targeted or Targeted The aim of un-targeted adversarial audio is to make an ASR model incorrectly transcribe speech while sounding similar to original input, while the aim of targeted adversarial attack is to cause an ASR model to output a specific transcription (target) injected by an adversary. This paper focuses on un-targeted adversarial attack.

2. Whitebox or Blackbox Threat Model In a whitebox threat model, the adversary assumes knowledge of the internal structure of the ASR model, while in a blackbox threat model, the adversary can only probe the ASR with input audio and analyze the resulting transcription. We use a blackbox threat model.

Most existing methods [85, 3, 26, 86] for ASR adversarial attack generation are *targeted and whitebox*. These methods suffer from one or more of the following drawbacks (1) Whitebox assumption is not practical and lacks portability since commercial ASR application developers do not typically reveal the internal workings of their systems, (2) time taken to generate attacks is considerable and cannot be used in real-time. , and (3) poor quality audio in attacks makes them easily detectable by defense techniques like [3, 87]. Existing few methods [88, 27] for *blackbox, targeted* attacks suffer from the drawback of intractable number of queries to the ASR, that are time-consuming and impractical. *Blackbox untargeted* attacks that do not rely on the knowledge of the internal NN structure or queries to the ASR would address the above limitations and the only known technique was proposed by Abdullah et al. in 2020 [2]. To create adversarial audio, they decompose the original audio and remove components with low-amplitude that they believe will not affect audio comprehension. Although interesting, their approach does not strive to ensure the adversarial and original audio sound similar. Additionally, difference achieved in transcribed texts is not measured or reported. We found the ability of their attacks in bypassing a state of the art defense system was not effective.

Proposed Attack Generation We propose a blackbox un-targeted attack generation approach that is faster, more portable across ASR, and robust to a state-of-the-art defense than Abdullah et al. Our framework, SPAT, for attacking ASR uses a psychoacoustics concept called frequency masking that determines how sounds interfere and mask each other. We manipulate masked (or inaudible) components of the original audio in such a way that their spectral density is different but they remain masked. Such a manipulation ensures the adversarial attack is indistinguishable from the original but has the potential to change the resulting transcription.

We propose three attack generation approaches centered around this idea – `Griffin Lim Reconstruction (GL)`,`Original Phase (OP)` and `Deletion (DE)`. Additionally, to help increase similarity to the original audio, we provide the option of selectively introducing perturbations to a small fraction of audio frames rather than all of them. The SPAT framework provides three frame selection options – `Random, Important` and `All`. Among them, the `Important` option identifies the frames that cause the most change to output text when set to zero and we then introduce perturbations to just these important frames.

We evaluate SPAT on three different ASR – Deepspeech [28], Sphinx [29] and Google cloud speech-to-text API, using two different input audio datasets – Librispeech [89] and Commonvoice [90]. We assess the effectiveness of our approaches for attack generation and frame selection using the metrics - `WER, Similarity`, attack `Success Rate` and `Detection score`. We also compare SPAT with a targeted whitebox state-of-the-art (SOTA) method [3] and an untargeted blackbox SOTA method [2]. It is worth noting that the scale of our evaluation is much bigger than existing work [3, 26, 2] as we use different audio datasets and ASR. We find our approach using `OP` or `DE` for attack generation combined with `Important` or `All` frame selection was effective at attacking all three ASR. Our techniques were $312\times$ faster than the whitebox targeted SOTA, and $7\times$ faster than blackbox targeted SOTA method. The defense system, Waveguard [4], was less effective at detecting attacks generated with our techniques compared with the other two SOTA methods.

In summary, the contributions in this chapter are as follows:

1. A novel approach and framework, SPAT, for untargeted blackbox adversarial attack generation on ASR based on frequency masking.
2. Frame selection option to selectively perturb frames in an audio.
3. Extensive empirical evaluation of the attack generation and frame selection options within SPAT on three ASR and two audio datasets. We also compare performance against SOTA whitebox and blackbox techniques.

3.2 Background

As stated in the Chapter 2, most current ASR comprise the following stages when transcribing an input audio to a text output: 1. Pre-processing to remove noise and detect human voice in the input audio, 2. Signal processing stage to extract audio features as Mel Frequency Cepstral Coefficient (MFCC) and 3. Prediction that uses the MFCC features from the audio to predict a probability distribution of characters for every time step or audio frame. From the character sequence distributions, an output selection algorithm, such as Beam search, is used

to select the most likely translated text. The process of MFCC extraction and the fundamental concepts of Automatic Speech Recognition (ASR) have been introduced in the thesis-level background chapter and will not be repeated here. We next present a brief description of the frequency masking concept used in SPAT.

3.2.1 Frequency Masking and Masking Threshold Computation

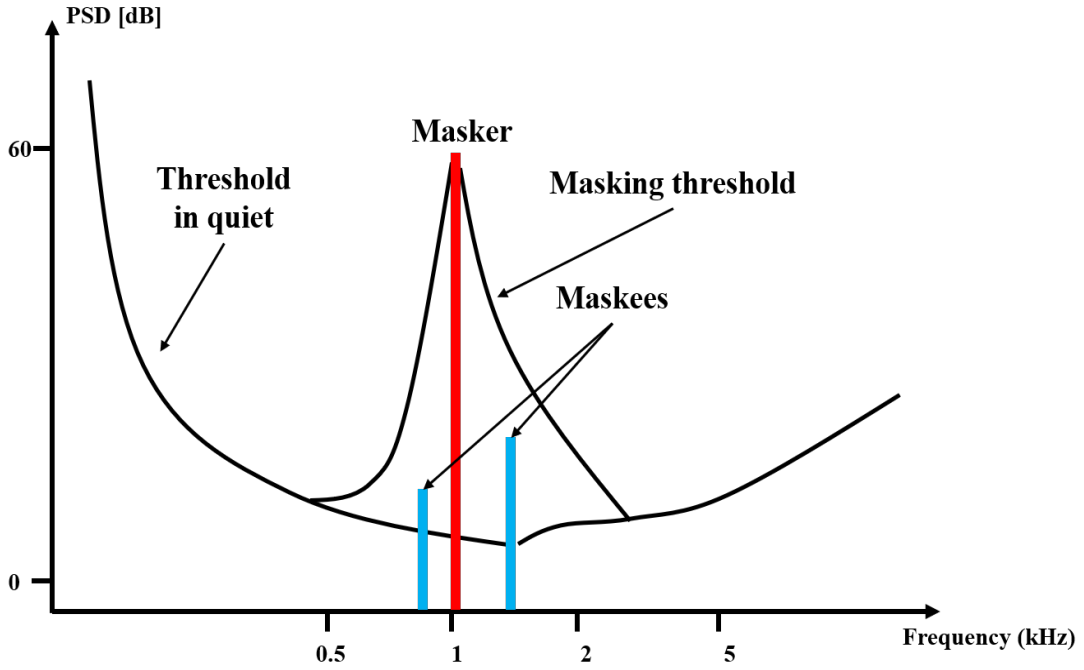


Figure 3.1: Frequency masking phenomenon: the masker creates a *masking threshold* in the nearby frequency domain such that other sounds below this threshold cannot be heard[1].

Frequency masking is a psychoacoustic phenomenon that occurs when the perception of a sound is affected and masked by the presence of another sound, distracting the ear from being able to clearly perceive the simultaneous sounds [91]. For example, on a quiet night, consider that the sound of chirping crickets is audible but in the presence of the TV sound, we stop hearing the crickets chirping as the TV sound masks it. In Figure 3.1, the TV sound would be the *masker* (seen as a red bar) that creates a masking threshold [91] which is the minimum level at which other sounds in the same frequency frame can be heard. The chirping sound of the crickets falls below the masking threshold (seen as a blue bar) and therefore is not audible in the presence of the TV. The chirping sound in Figure 3.1 would be the *maskee*.

Masking Threshold Computation To calculate the masking threshold for a given audio, we need to first convert the audio from the expression in the time domain to the frequency domain (using Fast Fourier Transform), then discard the phase information in the spectrum. We then use the amplitude information of the spectrum to calculate the log-magnitude power spectral density (PSD) of this audio. The PSD characterizes the energy distribution on a unit frequency, and is used widely to describe the frequency domain results of the signal [1, 91]. The red and blue bars in Figure 3.1 represent the PSD (in dB) of maskers and maskees, respectively, for the given frequency bin. According to [91, 26], maskers are identified from the audio PSD using two conditions: the PSD of a masker should be greater than the absolute threshold of hearing (ATH), and it must be the highest PSD estimate within a certain surrounding frequency range. After identifying the maskers, their respective masking thresholds will be computed using a two-slope function, described in [1]. If there are several maskers and associated masking thresholds, they will be combined into a global masking threshold for the audio like in [26]. Once the maskers are identified, the other PSDs in the audio are labelled maskees. A more detailed description of the computation of masker, maskee and masking threshold can be found in [1, 26].

We use this masking phenomenon observed with simultaneous sounds to create adversarial audio that sounds similar to the original audio but has the potential to produce a different transcription. We achieve this by first taking the original audio that is composed of many sounds, identifying the maskers and maskees in it using the approach from [1, 26] (red and blue bars in Figure 3.1). We then manipulate the PSD of the maskees so it stays below the masking threshold, ensuring they are not audible, like in the original audio. Nevertheless, this manipulation can still affect the transcribed text. We create the adversarial audio by composing together the unchanged maskers and manipulated maskees. In terms of our earlier example with the TV sound and crickets chirping, we identify the TV sound as the masker and the chirping crickets as the maskee. We then manipulate the PSD of the cricket sound, staying within the masking threshold, to produce an adversarial audio that composes the TV sound with the manipulated chirping sound. Section 3.4 describes the SPAT framework and the techniques used for manipulation in detail.

3.2.2 Griffin-Lim Algorithm

To construct an adversarial audio from the maskers and manipulated maskees in the amplitude spectrum, we use the Griffin-Lim (GL) algorithm that helps reconstruct audio waveforms with a known amplitude spectrum but an unknown phase spectrum[92]. The Griffin-Lim (GL) algorithm is chosen because it remains one of the most widely used and accessible phase-reconstruction methods in speech processing. When adversarial perturbations are introduced at the spectrogram level, phase recovery is required to reconstruct the waveform. GL provides a standard baseline for this purpose: it is computationally efficient, model-agnostic, and has

been extensively adopted in both classical and modern ASR pipelines. Steps in the algorithm are as follows: (1) Randomly initialize a phase spectrum, (2) Use this phase spectrum and the known amplitude spectrum to synthesize a new waveform through Inverse Short-Time Fourier Transform (3) Use the synthesized speech to get new amplitude spectrum and new phase spectrum through Short-time Fourier Transform, (4) Discard the new amplitude spectrum, (5) Repeat steps 2, 3, 4 for a fixed number of iterations. Output is a waveform with an estimated phase spectrum and the known input amplitude spectrum.

3.3 Related Work

Attack Type	Existing work
Whitebox-Targeted	Vaidya et al. [93], Carlini et al, [85, 3], Qin et al. [26],Yuan et al. [22],Yakura et al. [86], Schönherr et al. [94, 95], Szurley et al. [96]
Blackbox-Targeted	Zhang et al. [97], Alzantot et al. [88], Taori et al. [27]
Blackbox-Untargeted	Abdullah et al. [2]

Table 3.1: Existing work on adversarial ASR attack generations.

As mentioned in Section 3.1, existing adversarial attack generation on ASR models can be classified along two dimensions: 1. Targeted for a given transcription or untargeted, and 2. Whitebox, with knowledge of the internal ASR structure or Blackbox. Table 3.1 lists the existing techniques using these two dimensions and they are discussed in more detail in the rest of this Section.

3.3.1 Targeted Attacks

Vaidya et al. [93] pioneered the first whitebox targeted method for attacking ASR in 2015. Given the transcription to target, they gradually approach the target by continuously fine-tuning the parameters of the extracted MFCC features. Once the goal is reached, they use the obtained adversarial MFCC features to reconstruct the speech waveform. On the basis of Vaidya's work and in an effort to improve the efficiency of their approach, Carlini et al.[85] proposed Hidden Voice Command in 2016, adding noise that is often encountered in real life. However, neither of these two types of attacks can conceal the existence of noise, and such adversarial attacks can be easily detected as noise rather than effective commands.

Yuan et al. [22] proposed a method for embedding commands into songs so that when these songs are played, the commands will be translated by an ASR. Additionally, they improve the realistic nature of adversarial attacks by introducing noise generated by hardware devices. This approach, however, is restricted to songs as the carrier of commands, and is, therefore, limited in application scenarios.

Carlini et al. [3] in 2018 used a whitebox approach that applies gradient descent to modify the original audio so that the difference between the transcription and the target text is smaller. Their experimental results show their attack `Success Rates` reached 100% on Deepspeech ASR. However, their approach faces the following drawbacks: First, it can take up to several hours to generate attacks; second, the gradient descent method requires the attacker to have a good understanding of all the internal parameters and structures of the attacked system before it can be used; and finally the adversarial attacks generated will be invalid over other ASR.

Yakura et al. [86] proposed some improvements to [3] to maintain attack performance under over-the-air conditions (mixed with sound of the surrounding environment). They generate adversarial attacks accounting for noise caused by echo and recording in real life, so as to obtain more robust adversarial attacks. However, other shortcomings in Carlini et al.[3] (such as long generation time and weak transferability) have not been addressed.

In 2018, Schönherr et al. [94] developed a whitebox approach that applies the knowledge of masking threshold to generate adversarial attacks. They proposed to limit the generated noise below the masking threshold of the original audio to ensure that the obtained perturbation is not audible to the human ear. In more recent work [95], they introduced room impulse response (RIR) simulator to improve the robustness of examples that produces different types of noise for different environment configurations.

Inspired by Schönherr et al., Qin and Carlini et al. [26] developed a whitebox method and optimized perturbations to make it lower than the masking threshold of the original audio. This method achieved a 100% attack `Success Rate` on the Lingvo system. However, their algorithms only study the attack on traditional signal-processing-based ASR, and has not studied the end-to-end ASR that have emerged in recent years. Adbullah et al [98] adapted the algorithm for end-to-end ASR on the basis of those two. But again, it's also a targeted white-box attack, and those limitations are still unresolved.

Like other whitebox targeted approaches, their work lacks portability to other ASR and is time consuming for attack generation.

Around the same time, Szurley et al. [96] proposed a whitebox method similar to Schönherr et al. [94, 95] and Carlini et al. [26, 3] that constructed an optimization based on masking threshold and combined it with room reverberation. Their method reached a 100% `Success Rate` on Deepspeech but still suffers from limitations of lack of portability and time consuming attack generation.

Blackbox-targeted approaches Few Blackbox Targeted adversarial attack generation techniques exist in the literature [97, 88, 27]. Zhang et al. [97] in 2017 modulated the voice on the ultrasonic carrier to insert preset commands(like "Open the window") into the original audio. However, this method is not easy to reproduce as it uses hardware characteristics of the microphone to complete the attack. Alzantot et al. [88] proposed a iterative optimization method that adds a small amount of noise iteratively to a benign example until the ASR outputs a target label. Taori et al. [27] used a genetic algorithm to achieve iterative optimization, mutating benign examples until the ASR output matches a target label. These approaches for blackbox targeted attacks suffer from the following two weaknesses: First, they require thousands of queries to ASR to generate one adversarial attack, which is unrealistic. Secondly, these attacks are only applicable to ASR that aim to classify audios, not translate audios.

3.3.2 Untargeted Attacks

The only known untargeted blackbox adversarial ASR attack generation approach is that proposed by Abdullah et al. [2] in 2019. They construct an adversarial attack by decomposing and reconstructing the original audio. Specifically, they decompose the original audio into components called eigenvectors via Singular Spectrum Analysis (SSA). These eigenvectors represent the various trends and noises that make up the audio. They believe that eigenvectors with smaller eigenvalues convey limited information. They choose a threshold to classify eigenvalues as small and subsequently eliminate small eigenvectors. They then reconstruct an audio from the remaining components as the adversarial attack. We compare performance of our techniques against their approach in Section 3.6.4.

3.4 Methodology

In this section, we propose techniques for generating adversarial attacks for ASR. As seen in Figure 3.2, our framework, SPAT, has two important stages, 1. Audio Frame Selection and 2. Attack Generation. The general workflow in SPAT is as follows: Given an input audio example, we first select frames within it using one of the three techniques for audio frame selection – `Random`, `Important` and `All`. Independently, we generate manipulated audio from the

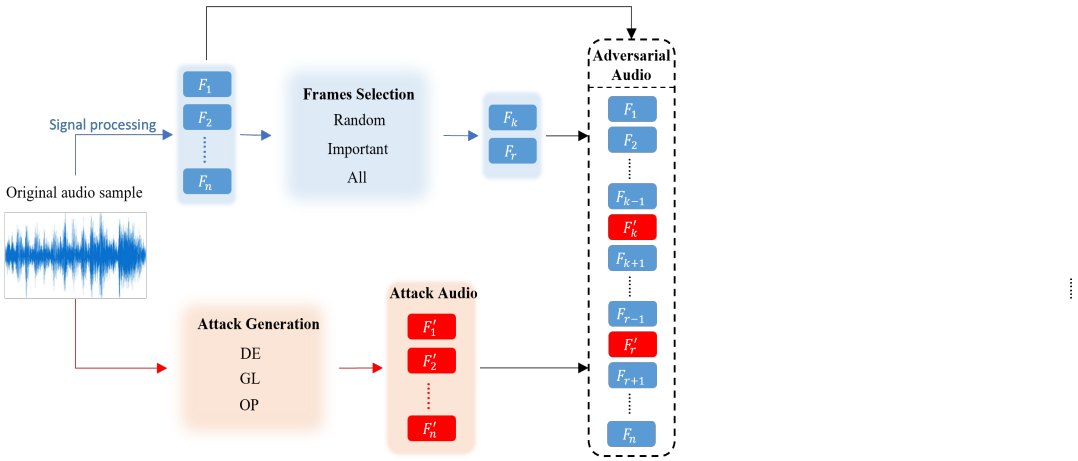


Figure 3.2: Our framework, SPAT, for generating adversarial attacks comprises of three stages, 1. Frame Selection, 2. Attack generation and finally 3. Adversarial audio formed by combining information in the first two stages.

input audio using one of three attack techniques – `GL Reconstruction (GL)`, `Original Phase (OP)`, `Deletion (DE)`. We then replace the selected frames in the original audio with corresponding manipulated audio frames while keeping the rest of the audio unchanged. The combination of original and manipulated audio frames forms the adversarial attack audio.

Threat Model and Assumptions The attack techniques in SPAT assume a black-box threat model, in which an adversary has no knowledge of the internal workings or architecture of the target ASR model. We treat the ASR as a black-box to which we make requests in the form of input audio and receive responses in the form of transcriptions in text format. We also assume that an adversary can only make a limited number of requests to the target ASR. We also accommodate the scenario when the adversary cannot make any requests to the target ASR (with `All` frames selected). Finally, we assume an over the line attack. This means that digital files are sent directly to the target ASR system for transcription, as opposed to playing back audio files over the air through speakers.

3.4.1 Stage 1: Frame Selection

We explore generation of adversarial audio by modifying a subset of frames in the entire audio. We provide three approaches to select audio frames that will be later manipulated – `Random`, `Important` and `All`. We will start by describing the technique to select `Important` frames.

Important:

The rationale for selecting important frames is to restrict manipulation to a small number of significant frames. This allows the adversarial audio to remain similar to the original while still affecting the output transcription text. We define importance of frames based on the proportion of Word Error Rate (`WER`[2]) produced by masking that frame in the original audio. The steps involved in selecting important frames are as follows,

1. For every input audio example, record output transcription.
2. Pick one of the input audio examples. For every frame in the processed audio example, set it to zero (masked) while keeping the remaining frames unchanged. Record translated text using the ASR for the masked audio.
3. Compute `WER` between the masked and original output. Repeat this for all frames. The frames that result in a non-zero `WER` are identified as important frames for that audio example. Magnitude of `WER` change for frame selection can be altered to suit needs.
4. Repeat Steps 2 and 3 for the remaining input audio examples.

At the end of this process, every input audio example is associated with a list of important frames.

---

2. `WER` is a common metric to evaluate the difference in ASR transcription between original versus adversarial audio. The formula is provided in Sec 3.5.2

Random:

To enable us to compare the effectiveness of only using important frames in frame selection, we also provide a means to select frames randomly. The number of frames selected for a given audio example is set to be the same as the number of important frames in that audio.

All:

We simply use *all* the frames from the manipulated audio generated in Stage 2 (see Section 3.4.2). Using `All` frames helps us assess how much `WER` was achievable. In addition it helps quantify the tradeoff in `WER` and `Similarity` when compared to frame selection with `Important` and `Random`. It is worth noting that using `All` frames requires *no* queries to the ASR. Therefore, if the threat model assumes no queries then we would select `All` frames in Stage 1 of our approach.

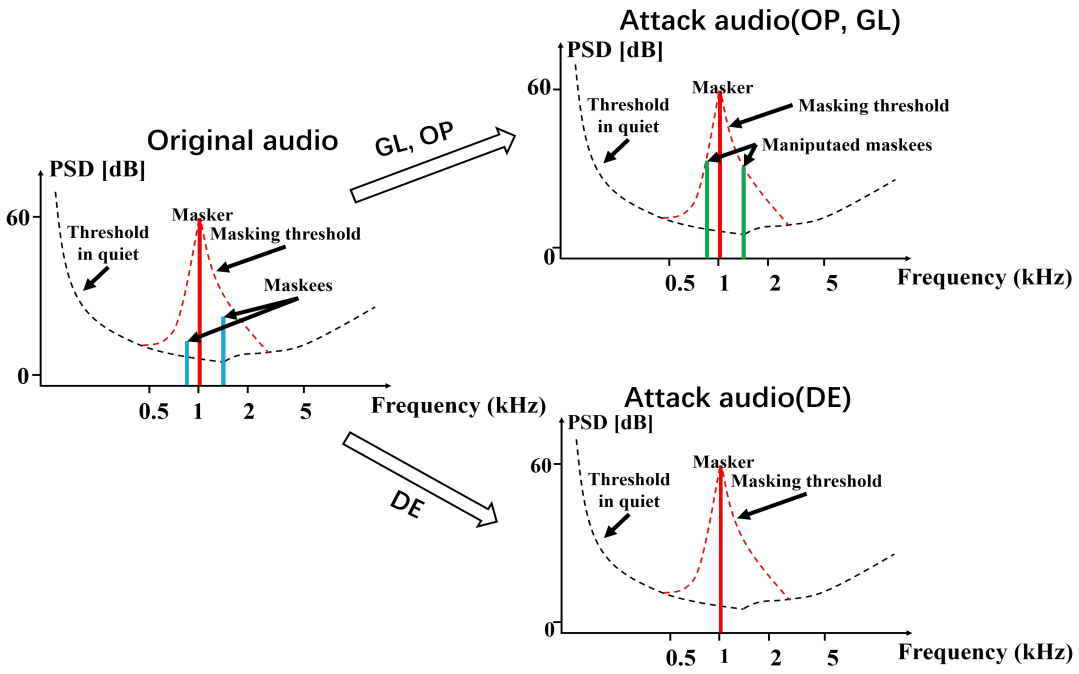


Figure 3.3: Attack generation methods, GL and OP, increase the PSD of maskees to the masking threshold. Attack generation with DE suppresses the PSD of maskees to zero.

3.4.2 Stage 2: Attack Generation

We discuss three attack generation techniques within SPAT – GL, OP and DE, that manipulate the amplitude spectrum of the input audio example using the concept of frequency masking, described in Section 3.2.1. We illustrate the manipulations in Figure 3.3 and describe them in the Sections below. All three techniques take the input audio, generate audio frames in the frequency domain (obtained with sampling and fast fourier transform), with each frame having amplitude and phase information. For each frame, we compute the masking threshold, maskers and maskees using established techniques discussed in Section 3.2.1

GL Reconstruction (GL)

As seen in the top part of Figure 3.3, `GL` (and `OP`) increases the PSD of all maskees (blue bars in the original audio) to the global masking threshold. Masker PSDs remain unchanged. We then compute an updated amplitude based on the maskers and altered maskees PSD inversely [1][3]. `GL` discards phase information of the input audio waveform. Instead, it estimates phase information using the GL reconstruction technique discussed in Section 3.2.2. The estimated phase information is combined with the updated amplitude information and is used to synthesize the attack audio through inverse FFT.

Original Phase (OP)

The primary difference between the `OP` and `GL` technique is in the phase information. Estimating phase using the GL algorithm introduces distortion and lack of consistency across multiple runs. To avoid this problem, the `OP` technique retains phase information from the original audio. We believe using phase information from the original audio to synthesize the attack audio will make it more similar to the original audio.

Deletion (DE)

Previous methods, `OP` and `GL`, ensure the attack audio sounds no different from the original input by increasing the PSD of the maskees up to the maximum limit (which is the masking threshold) for them to remain masked. The `DE` technique, on the other hand, suppresses the PSD of the maskees to the minimum value of zero which is akin to deleting them. This manipulation will not affect the audio perception as the masking threshold is unaffected. The `DE` technique, thus, deletes all maskee PSDs that are hidden under the masking threshold. Subsequently, we use the modified amplitude after deletion and combine it with the *original phase* information from the input audio (similar to `OP`'s use of phase). We use inverse FFT as before to synthesize attack audio from the amplitude and phase information.

---

3. $Amplitude(k) = N\sqrt{10^{\frac{PSD(k))}{10}}}$, where k is the index of the frequency bin and N represents the length of frame.

3.4.3 Stage 3: Combining Original and Attack Audio

In this final stage, we create an adversarial attack by taking the original audio, replacing the selected frames (identified in Stage 1) with corresponding frames from the attack audio (generated in Stage 2). Other frames from the original audio are left unchanged. This modified version of the original serves as an adversarial attack.

3.5 Experiments

We evaluate the effectiveness of our techniques within SPAT, described in Section 3.4, using two different datasets – (1) 200 audio samples from Librispeech [89] and (2) 200 audio samples from Commonvoice [90]. We use three ASR in our evaluation, namely, Deepspeech [28], Sphinx [29], and Google ASR. Our choice of datasets and ASR were inspired by their use in related work for adversarial ASR attack generation [2][3][26][97]. We discuss the defense system used to assess the effectiveness of the adversarial attacks, evaluation metrics and the research questions in our experiments in the rest of this Section.

3.5.1 Detection and defense

The ability to evade defense systems is an important measure of effectiveness for adversarial attacks. Defense systems have evolved to detect and defend a significant fraction of adversarial attacks.

In our experiments, we use a SOTA adversarial audio detection and defense system, Waveguard [4], proposed by Hussain et al. We chose Waveguard as our defense system as it is demonstrated to be faster, more effective and capable of detecting both targeted and untargeted attacks compared to existing detection techniques, like Temporal Dependency Detection Method [99]. We report how well Waveguard performed (as an AUC score) in detecting adversarial attacks in our experiments.

Attack detection with Waveguard is divided into two steps. The first step is to transform the input audio using one of several functions that are meant to preserve (or closely preserve) the transcription text. For example, one of their transformations – Mel Spectrogram Extraction and Inversion – first extracts MFCC features from input audio and reconstructs the audio from MFCC features. The second step is to compare the Character Error Rate(CER) between the transcription text for the original and transformed audio. If the difference between the texts is greater than a predefined threshold, then the input audio is classified as adversarial, and benign otherwise.

3.5.2 Evaluation Metrics

We use four metrics to measure the effectiveness of our techniques – `Word Error Rate (WER)`, `Similarity`, `Success Rate` and `Detection score`. We are interested in generating adversarial attacks that sound similar to the the original audio (high `Similarity`) but produce a transcription different from the original (high `WER`). Additionally, we would like the technique to be portable, i.e generate adversarial attacks that are usable across several ASR (high `Success Rate`). Finally, we want the generated attacks to be robust to get past SOTA defense systems, like Waveguard [4] (lower `Detection score`).

We provide definitions of each of these metrics below.

`WER` is a common metric to evaluate the difference in ASR transcription from original versus adversarial audio [100] [101]. `WER` is computed using Equation (3.1),

$$\text{WER} = \frac{\text{Insertions} + \text{Substitutions} + \text{Deletions}}{\text{Total Words in Correct Transcript}} \tag{3.1}$$

Similarity We use the widely used `PESQ` metric [102] that measures quality of audio relative to a reference audio to assess similarity of adversarial audio to the original. The `PESQ` algorithm accepts a noisy signal, which in our case is the adversarial attack, and an original reference signal, which is the input audio for our method. The `PESQ` score ranges from -0.5 to 4.5. The higher the score, the better the voice quality. According to [103], audio quality is deemed “good" when its `PESQ` score is above 3.0. We use this standard for classifying the quality of the adversarial audio. In this paper, we use `Similarity` metric to mean the `PESQ` score.

`Success Rate` shown in Equation (3.2), refers to the ratio of adversarial attacks that can successfully attack a given ASR. A successful attack, as defined by Abdullah et al [2], happens when the adversarial attack results in a non-zero WER with respect to the original transcription.

$$\text{Success Rate} = \frac{\text{Number of successful attacks}}{\text{Total number of adversarial attacks}} \tag{3.2}$$

`Detection score` refers to the effectiveness of the Waveguard defense system in correctly classifying adversarial attacks. We use the area under the curve (AUC) metric, reported by Waveguard [4], to evaluate correct classification of adversarial attacks. The AUC score ranges from 0.0 to 1.0. We aim for a lower Waveguard AUC score or `Detection score` with our techniques.

3.5.3 Research Questions

We aim to answer the following research questions (RQs) in our experiments,
RQ1: Which frame selection method in SPAT among `Random, Important, All` performs best?
We compare the `WER` and `Similarity` achieved by the different frame selection techniques across three different ASR and two input audio datasets. Answering this research question will help us assess the value of selecting a subset of frames versus just changing the whole audio.

RQ2: Which attack generation technique among `GL, OP, DE` performs best?
We compare the `WER`, `Similarity` achieved by the different attack generation techniques across three different ASR and two different input datasets. We also measure `Time` taken by each technique.

RQ3: Are the adversarial attacks portable across ASR?
One of the primary selling points of our techniques is that they are blackbox and untargeted, and therefore agnostic to the structure and workings within ASR. We validate this by evaluating the `Success Rate` of the generated adversarial attacks across three different ASR.

RQ4: Do SPAT generated attacks perform better than SOTA techniques?
We selected representative and high-performing SOTAs in our comparison, namely a whitebox targeted technique proposed by Carlini et al [3], and a blackbox technique by Abdullah et al [2].

Carlini et al. generate adversarial attacks using Deepspeech ASR and the Commonvoice input dataset. To allow comparison, we use the same ASR and input dataset with our techniques. Owing to the targeted nature of their technique, they require the transcription text to be specified in advance. To address this need, we use the transcription from Deepspeech ASR with adversarial attacks generated by our technique as Carlini et al.'s target. We then compare our technique with Carlini et al. with respect to time taken to generate adversarial attacks, `Similarity` to original audio, `Success Rate` on other ASR, Google and Sphinx, and `Detection score`. Since the transcription text in both techniques are the same, it is not useful to compare `WER`.

We compare our technique against Abdullah et al. using `WER, Similarity, Success Rate, Detection Score, Time` over different ASR and both the Commonvoice and Librispeech dataset.

Experiment settings We use Google Colab Pro with two NVIDIA Tesla T4 GPUs(16GB RAM, 2560 cores) to run our experiments. We use the following audio parameters in our experiments: Sampling rate of $16000HZ$, frame length of 2048 and frame shift of 512.

3.6 Results and Analysis

We present and discuss the results from our experiments in the context of the research questions presented earlier.

It is worth noting that `WER` and `Similarity` are measured for each attack, while `Success rate` and `Detection score` are measured across an entire dataset. Techniques should try to maximise `WER, Similarity` and `Success rate` while minimising `Detection score` by Waveguard.

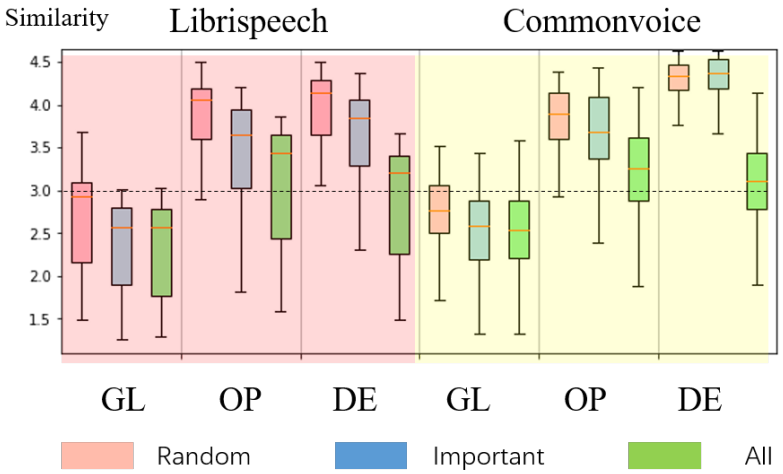


Figure 3.4: Box plots of the `Similarity` of the adversarial attacks generated with all datasets.

3.6.1 RQ1: Comparison of Frame Selection Techniques

The best performing frame selection technique is one that achieves high `WER` and high `Similarity` across audio examples. However, these two metrics are often conflicting. We discuss and compare `WER` and `Similarity` achieved by the three frame selection techniques in SPAT below. Figures in Table 3.2 shows the `WER` achieved by different frame section techniques for the Librispeech and Commonvoice datasets across different ASR and attack generation techniques while Figure 3.4 shows the `Similarity` achieved.

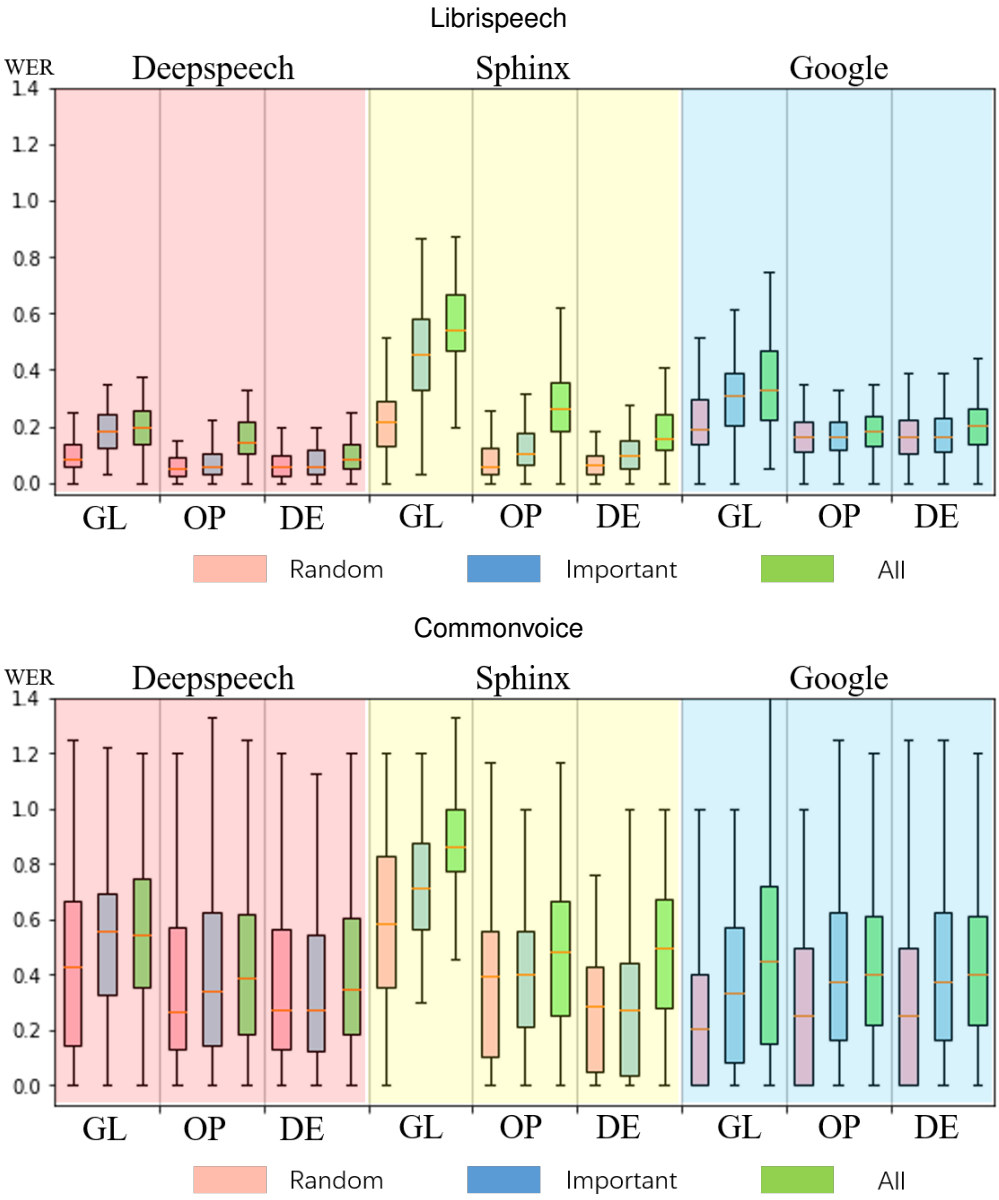


Table 3.2: Box plots of the `WER` of the adversarial attacks generated with two different datasets.

	Librispeech			Commonvoice		
	GL	OP	DE	GL	OP	DE
Deepspeech	96%	95%	91%	95%	90%	90%
Sphinx	99%	96.5%	94%	98%	89%	90%
Google	99%	97.5%	95.5%	85%	80%	80%
Average	98%	96.3%	93.5%	92%	86.3%	86.6%

Table 3.3: The `Success Rates` of the adversarial attacks with `GL,OP,DE` attack generation methods across the three ASR and two datasets.`All` frames is used as the frame selection method.

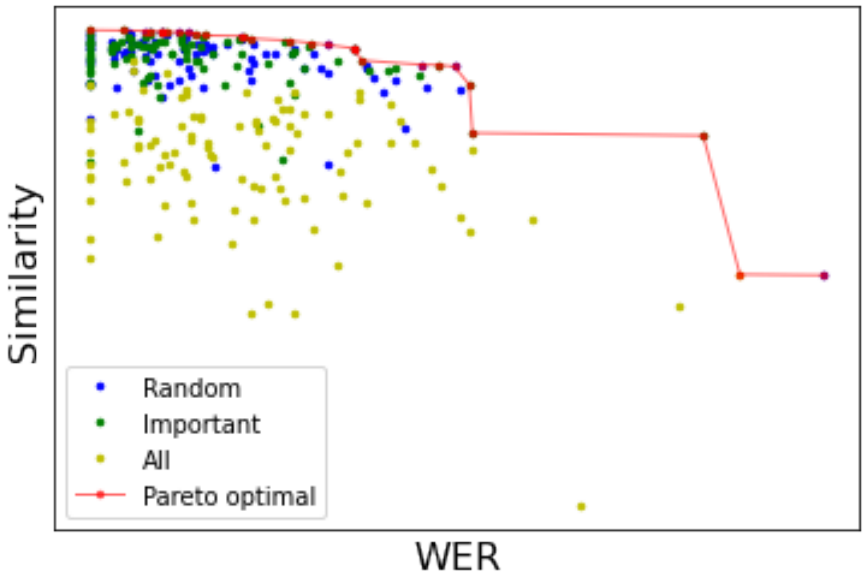


Figure 3.5: Pareto front over adversarial attacks generated by `Random`, `Important` and `All` frame selection techniques on Commonvoice dataset and Deepspeech ASR using `DE`.

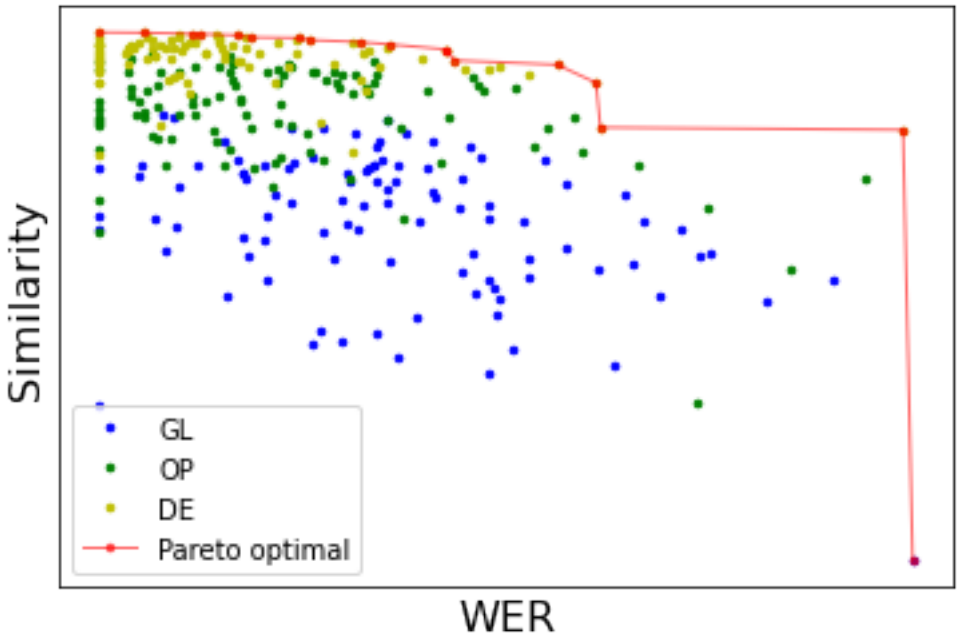


Figure 3.6: Pareto front over adversarial attacks generated by `GL`, `OP` and `DE` on Commonvoice dataset and Deepspeech ASR using `Important` frames.

Technique	Time	Similarity	Success rate			WER			Detection score
			Deepspeech	Sphinx	Google	Deepspeech	Sphinx	Google	
Carlini [3]	780 seconds	3.63	N/A	77%	33%	N/A	N/A	N/A	0.67
Abdullah [2]	18 seconds	3.12	80%	77%	54%	0.39	0.44	0.14	0.65
OP	3.5 seconds	3.65	90%	89%	80%	0.46	0.47	0.40	0.52
DE	2.5 seconds	4.29	90%	90%	80%	0.45	0.50	0.38	0.55

Table 3.4: Comparison of `OP,DE` with Abdullah et al. [2] and Carlini et al. [3] with respect to generation time for per adversarial attack, `Similarity` to original audio examples,`WER`, `Success Rate` and `Detection score` against defense system [4] in attacking all three ASR

`All` frames We find in Table 3.2 and Figure 3.4, that the `All` frame selection achieves the highest `WER` and lowest `Similarity` compared to `Important` and `Random` across ASR, input datasets and attack generation methods. This is in line with our expectations as the other two frame selection techniques select a small part of the audio to introduce noise into achieving lower `WER` but higher `Similarity` to original audio.

`Important` versus `Random`: For most combinations of ASR, dataset and attack generation, we find `Random` frame selection produces the lowest `WER` and the highest `Similarity`, while `Important` frame selection results in a `WER` and `Similarity` between `Random` and `All`.

Statistical Analysis. We confirmed the statistical significance (at 5% significance level) of the difference in means between the frame selection techniques using one-way Anova and did a post-hoc Tukey's Honest Significant Difference (HSD) test to reveal which differences between pairs of means are significant. The detailed P-values for pairwise comparisons of `WERs` and `Similarities` between frame selection techniques are provided in Appendix A.1[4]. For the `WER` metric, we find the `All` frames selection technology is significantly better than `Important` and `Random` on majority of ASR, dataset, attack technique combinations. In contrast, for `Similarity` measure, `Random` and `Important` frame selections significantly outperformed `All`.

Pareto front Owing to the conflicting nature of the `WER` and `Similarity` metrics, all three frame selection techniques achieve a trade-off between them. We use the Pareto front with these two metrics, shown in Figure 3.5 for one of the datasets and ASR, to determine the number of non-dominated attack examples (that fall on the Pareto front) from each frame selection. We find `Important` frame selection has the most number of non-dominated attacks (17 examples); `Random` was second with 12 examples, while `All` frames only had 1 non-

4. An additional extension file is also available at https://anonymous.4open.science/r/lalalala-9DEE/apsec2022_extension.pdf

dominated attack example. This trend is observed across all ASR, attack technologies and datasets (see results in Extension file Section 1.1.3). Based on the number of non-dominated examples, we believe that Important frames is effective at achieving a trade-off between WER and Similarity.

Summary In terms of WER, we find All frames performs best. However, Important and Random frames perform better in terms of Similarity. We find Important is the best at optimising trade-off between the two metrics, achieving reasonable performance in both WER and Similarity.

3.6.2 RQ2: Comparison of Attack Generation Techniques

We present WER achieved by GL, OP, DE using different ASR and datasets in Table 3.2, while we show Similarity achieved in Figure 3.4. Best performing attack generation technique is one that results in a high WER and high Similarity to original audio.

WER Performance GL attack generation performs better than both OP and DE in terms of WER achieved. We confirm the differences are significant using One-way Anova and Tukey's HSD test (The corresponding P-values are reported in Appendix A.2.). Between OP and DE attacks, OP outperforms DE with DeepSpeech and Sphinx ASR over the Librispeech dataset. There is no significant difference between the two techniques over the other dataset and ASR.

Similarity Performance Both OP and DE significantly outperform GL in terms of Similarity, confirmed with pairwise comparison using one-way Anova followed by Tukey's HSD test (The corresponding P-values are reported in Appendix A.2.). The median Similarity or PESQ score for GL tends to be below the value of 3.0 (shown by the dashed line), irrespective of frame selection used. According to Beuran et al. [103], the standard for good quality audio is a PESQ score of greater than 3 and GL technique does not meet this standard in our experiments. We believe this is because GL uses estimated, rather than actual, phase information which causes distortion that reduces the PESQ score.

Between OP and DE, there is no significant difference in their Similarity performance. The benefit with using DE lies in faster generation of an adversarial attack. The average time to generate a single adversarial attack using DE is $2.5 seconds$, a second faster than the OP technique ($3.5 seconds$ on average) as OP relies on calculating the masking threshold for every input example.

Pareto Front As with RQ1, we draw the Pareto front using WER and Similarity, shown in Figure 3.6. For Deepspeech ASR using Important frames in Figure 3.6, we find DE technique has the most number of non-dominated attacks (16 examples); OP is second with 5 examples, while GL only has 2 non-dominated attack example. This trend is observed across most ASR, frame selections and datasets. However, for the All frame selection, OP has the most number of non-dominated attacks on Deepspeech and Google (Results available in Section 1.1.3 of the Extension file).

Summary Based on the number of non-dominated examples, we believe that when using Important and Random frame selection, DE is a suitable choice for optimising both WER and Similarity. When using All frame selection, OP is a better choice. Independently, DE is the fastest attack generation among the three techniques.

3.6.3 RQ3: Portability across ASR

We evaluate portability of the adversarial attacks generated by OP,GL,DE across the three ASR using the Success Rate metric, described in Section 3.5.2. Table 3.3 presents Success Rates achieved with the Librispeech and Commonvoice datasets.

We find GL achieves the best success rates over all ASR, with both the Librispeech dataset (average of 98%) and the Commonvoice dataset (average of 92%). OP comes next, performing better than DE on the Librispeech dataset (96% versus 93.5%, respectively). OP and DE have similar performance over the Commonvoice dataset (average of 86%).

Summary All three attack generation techniques have high success rates across the three ASR producing portable adversarial attacks. GL outperforms OP and DE in portability but the magnitude of difference is small (on average 2% to 5%). OP and DE have comparable performance on the ASR, especially with the Commonvoice dataset.

3.6.4 RQ4: Comparison to Existing Techniques

We compare performance of SPAT against a whitebox targeted technique proposed by Carlini et al. [3] and a blackbox untargeted technique proposed by Abdullah et al. [2] using the metrics – WER, Similarity, Success rate, Time, Detection score. Within SPAT, we use OP and DE for attack generation as they perform best in terms of Similarity and WER[5]

5. We use the best performance between All and Important frames.

Comparison with Carlini et al

We fix the ASR to Deepspeech and input dataset to Commonvoice to match the experiments in Carlini et al. [3]. We show results in Table A.18. We do not compare WER as the target text for Carlini et al. [3] is the transcription text from our adversarial attacks, so there will be no difference.

Time and Similarity We find time taken to generate attack examples is faster with our approaches, OP and DE, compared to Carlini et al. with a maximum speedup of $312\times$ achieved with DE. We also achieve higher Similarity scores – 4.3 (DE) and 3.7 (OP), compared to 3.6 by Carlini et al. We confirm the statistical significance (at 5% significance level) of the observed differences in Similarity using one-way Anova and Tukey's Honest Significant Difference (HSD) test. We find our techniques are a clear winner in terms of time taken, and outperform Carlini at al. in Similarity.

Success Rate To evaluate portability of adversarial attacks, we transcribe the adversarial attacks using Google and Sphinx (since DeepSpeech is used by Carlini et al.). We find when used with Google ASR, adversarial attacks generated by Carlini et al. have a much lower Success Rate than our techniques (33% versus 80%), respectively. For Sphinx, the difference in Success Rate is smaller but the trend remains (77% Carlini versus 89% to 90% for ours). The lower Success Rate observed with Carlini et al. is because their technique specifically targets the neural network inside Deepspeech, and may not be as effective when used on other ASR with different NNs. This is a drawback also encountered with other white-box attacks. However, since our method is blackbox, we find it is easier to port our adversarial attacks to different ASR.

Detection score Attack examples generated by Carlini et al. are more easily detected by Waveguard, with a higher Detection score score of 0.67, compared to techniques in SPAT, whose Detection scores are 0.52 for OP and 0.55 for DE. We believe this is because Carlini et al use noise in their attack generation which is detected more easily by Waveguard. We find SPAT attack generation with OP and DE performs better than Carlini et al at evading the Waveguard defense.

Across all four evaluation metrics, we find one of the two techniques from SPAT is the winner (highlighted in red in Table A.18), outperforming Carlini et al [3] across all metrics.

Comparison with Abdullah et al

Like SPAT, Abdullah et al. [2] use a blackbox, untargeted attack generation technique that is meant to be fast and portable on different ASR. Unlike the comparison with Carlini et al., we can include WER as a performance metric (in addition to the other 4 metrics) and Deepspeech ASR in our comparison. We discuss performance for each of the metrics below using the Commonvoice dataset[6].

Time and Similarity We find attack generation with OP and DE is much faster than Abdullah et al. ($5\times$ and $7\times$ faster, respectively). For the Similarity metric, SPAT outperforms Abdullah et al. with both OP and DE attack generation (at 5% significance level, P-value tables in the Appendix A.3.)

Success rate, WER and Detection score Attack examples generated with OP and DE have a higher Success rate than Abdullah et al. across all ASR. We see a similar trend with WER, where OP and DE outperform Abdullah et al. (at 5% statistical significance). Finally, OP and DE surpass Abdullah et al. with respect to getting past Waveguard's defense system by achieving lower detection scores of 0.52 and 0.55, respectively, versus 0.65 for Abdullah et al..

In summary, we find our attack techniques, OP and DE, surpass Abdullah et al. for each of the five evaluation metrics (best performing is highlighted in red in Table A.18).

Threats to Validity

There are three threats to validity in our experiments based on the selected ASR, speech datasets and the metrics used in evaluation.

Firstly, we only use three ASR among dozens of commercial and non-commercial ASR in our experiments to evaluate the effectiveness of our attacks. Results may vary on other ASR. However, our techniques are meant to be ASR agnostic so we believe they will be applicable to other ASR. It is worth noting that the number of ASR in our experiments is at par or exceeds that used in Section 3.3. We plan on conducting a more extensive evaluation in the future.

Secondly, we use audio samples from two common speech datasets – Librispeech [89] and Commonvoice [90]. The adversarial examples we generate are a manipulation of the input audio. It would be interesting to evaluate our technique on audio samples in other speech datasets. Given the time consuming nature of the experiments, the number of samples from the different attack generation techniques and their combination with frame selection techniques, we were unable to scale our experiments further.

---

6. Results for Librispeech dataset follow a similar trend and can be viewed in Extension file Section 1.3.2.

Thirdly, we use metrics `WER, PESQ score` for `Similarity`, `Success Rate` and attack `Detection Score` in our evaluation of adversarial examples. These metrics have been used separately in other related work [100, 101, 2, 102] which led to their selection in our experiments. We have also tried Cosine Similarity of MFCC features in place of PESQ score and the trends were similar between the different techniques in SPAT. The choice of metrics for evaluating adversarial examples in this field have not been standardised and there is a range of metrics across several papers. We have tried our best to capture several metrics in our evaluation to avoid bias along any one dimension.

There is a limitation observed in this study concerns the relationship between attack success and the perceived "realisticness" of the resulting transcription. High word error rates (WER) do not necessarily indicate that adversarial outputs are meaningful. In some cases, attacks led to nonsensical word sequences which counted as successful under WER-based evaluation but would be implausible in realistic scenarios. Conversely, other attacks produced transcriptions that remained semantically coherent, and thus more realistic from a user perspective, despite achieving lower WER.

This trade-off suggests that robustness evaluation should not rely solely on quantitative metrics such as WER, Similarity, or Success Rate. Instead, semantic plausibility must also be considered, for example through perceptual assessment or task-oriented evaluation, to capture how attacks would be experienced in practice.

3.7 Conclusion

This chapter has addressed **RQ1: How do Automatic Speech Recognition (ASR) systems behave under adversarial perturbations, and what does this reveal about their robustness in real-world deployment?** To answer this question, we introduced SPAT, a psychoacoustically motivated adversarial framework that generates perturbations which remain largely imperceptible to human listeners while substantially degrading recognition accuracy across both open-source and commercial ASR systems.

We proposed a blackbox untargeted adversarial attack generation technique for ASR using frequency masking to make the adversarial audio sound similar to the original while producing a change in the transcription. Our framework, SPAT, provides three attack generation options – `GL, OP` and `DE`. We also provide the option of selectively introducing perturbations to a small fraction of audio frames using three frame selection options — `Random, Important` and `All`. Evaluation of our techniques over three ASR and two audio datasets showed that our techniques can be effective at achieving high `WERs` (average of 44% with `OP+All`) while also achieving high `Similarity` (average of 3.93 with `OP+Important`). The choice in attack generation and frame selection helps achieve a good balance between these two metrics, with

DE attack generation and Important frames achieving the best trade-off. We also confirmed that our techniques were portable across ASR and superior to existing whitebox targeted technique [3] and blackbox untargeted technique [2] in terms of WER, Similarity, Success Rate, Time and Detection score.

Beyond these quantitative findings, the results highlight several important concerns for ASR developers. First, robustness cannot be inferred from performance on clean test data alone; systems that appear accurate in standard evaluations may still be highly vulnerable to adversarial manipulation. Second, the imperceptibility of the perturbations underscores the potential for real-world misuse: users may not detect that inputs have been manipulated, even though outputs are corrupted. Third, our analysis of the trade-off between attack success and perceptual realism suggests that evaluation should go beyond binary success rates and incorporate whether modified transcriptions retain semantic plausibility.

Taken together, this chapter demonstrates that adversarial robustness is a challenge for speech AI. Addressing RQ1, we show that current ASR systems remain fragile under imperceptible perturbations, reinforcing the need for adversarial evaluation and robust defence strategies in both research and deployment contexts.

Chapter 4

Explainability: Post-hoc Explanation on ASR

Having investigated the robustness of ASR systems under adversarial perturbations in the last chapter, we now turn to another core concern in speech AI: explainability. While robustness reveals how models behave under input manipulations, explainability addresses whether users can understand and assess the system's outputs. This transition is not only technical but also reflects broader concerns about the transparency and accountability of AI systems—issues that are increasingly emphasized in regulatory frameworks, as discussed in the Chapter 1.

In line with these considerations, this chapter explores post-hoc explainability in the context of ASR. We focus on realistic usage scenarios where users interact with commercial ASR APIs as black boxes, with no access to model internals. To support interpretation under such constraints, we apply a suite of post-hoc methods—LIME [34], SFL [35], and Causal [36]—that rely solely on input-output behavior. Our goal is to examine whether these tools can highlight the parts of the input audio responsible for a given transcription, thereby offering explainable insights into model decisions and supporting trust in ASR systems deployed in practice.

At the time of this work, explainability research in ASR largely involved direct adaptations of methods from vision and text, such as LIME. These methods could identify influential input regions but did not account for the temporal and acoustic structure of speech. The novelty of this chapter is in adapting perturbation-based attribution methods specifically for ASR, introducing speech-aware perturbation strategies and systematically analysing their behaviour. This represents one of the first targeted studies of post-hoc explainability in ASR, extending the scope of XAI beyond the visual and textual modalities where it was predominantly focused.

The structure of this chapter includes a brief introduction, methodology, experiments, results and analysis, and a concluding discussion.

4.1 Introduction

Given the prevalence of ASR in our daily lives, concerns over their quality and accountability have become particularly important. The complex and often opaque nature of neural network-based ASR systems [104] makes it challenging to ensure their quality. We focus on improving ASR quality assessment by offering explanations for specific transcriptions, which can enhance our understanding of the ASR and potentially help identify and correct the faults causing transcription failures, as well as support the accountability process. This is in line with Madsen et al. [37], who state, "*While some issues can be partially mitigated by robustness and fairness metrics, it is often impossible to consider all failure modes. Therefore, quality assessment should rely on model explanations.*"

XAI techniques[1] have seen rapid development over the past five years. These methods are widely applied in classification tasks, such as image classification [60, 61, 62, 34, 63, 36] in computer vision or sentiment analysis in natural language processing (NLP) [64]. For instance, in image classification, XAI methods identify the pixels most critical to the model's prediction. For a "cat" label, these pixels often highlight regions such as the eyes, whiskers, or ears. Similarly, in sentiment analysis, a sentence like "This exam is quite difficult," classified as "Negative," can be analyzed to determine the contribution of words like "difficult" and "quite" to the prediction. A more comprehensive coverage of existing techniques can be found in recently published surveys of XAI techniques [105, 106, 107, 108, 37]

In recent years, similar explanation techniques have also been adapted on speech-AI tasks. For example, Wu et al. [109] analyzed the importance of speaker attributes, such as gender and age, in speaker verification systems, while Ben et al. [110] also explored the role of hand-crafted features like F0 and format in speaker recognition tasks. These methods have also been applied in tasks like accent classification [111], emotion classification [112, 113], hate speech detection [114] and speech-based health diagnostics [115].

However, these existing techniques are largely confined to classification tasks, where an entire input sequence is associated with one single output label. For sequence-to-sequence models, where both the input and output are variable-length sequences, explanation techniques often focus on the internal mechanisms of neural networks to analyze how specific information is processed across layers, rather than identifying which parts of the input are most responsible for the output. In the field of NLP, research on large language models like GPT has explored the semantic roles of individual neurons at various layers [116, 117]. Similarly, in the context of ASR models, efforts have been made to uncover how these systems process speech information. For instance, [118] developed time-independent Neuron Activation Profiles (NAPs) to represent the activation patterns of neurons for specific input groups. Clustering these profiles

1. Our focus is specifically on *post-hoc* explainable methods, which provide explanations *after* the model has been trained.

has demonstrated that different types of input (e.g., phonemes, syllables) trigger distinct patterns across layers, highlighting how layers specialize in extracting certain speech features. Another study [119] aggregated feature maps after ReLU activations in each transposed convolutional layer, producing interpretable time-series representations that reveal the types of speech features encoded at each layer. This approach provided insights into how different speech characteristics, such as pitch or formants, are captured progressively through the network. Despite these advancements, such techniques face two key limitations. First, they are highly dependent on specific white-box models and neural network architectures, limiting their generalizability to other models. Second, they do not explain how the ASR system generates its variable-length transcription from the input speech.

Compared to classification tasks, transcriptions pose greater challenges for explanation due to two primary reasons. First, most interpretable machine learning techniques are inherently unsuitable for variable-length outputs. Second, while single-label classifications allow for straightforward correctness checks, evaluating transcription accuracy requires complex semantic comparisons, where minor deviations in wording may still yield acceptable results. Addressing these challenges requires new methods capable of generating meaningful explanations for transcription tasks without relying on the internal architecture of the ASR models.

To explain ASR transcriptions, we first propose a method to categorize them as either Correct or Incorrect by assessing the degree of similarity between the ASR output and the desired transcription. After this, we aim to provide an explanation for the transcription by identifying a subset of audio frames from the input that are directly responsible for the output. It's important to clarify that, in this context, *frames* refer to raw data segments along the time dimension and should not be mistaken for frames derived from features such as MFCCs in the frequency or cepstral domains. To provide explanations, we adapt existing XAI techniques from image classification - LIME [34] SFL [35], and Causal [36].

In this chapter, we conduct large-scale experiments using 1,000 samples from the Common-Voice dataset [90] and additional 1000 samples from the TIMIT dataset [39].

We evaluate the quality of explanations from different techniques in terms of their size and consistency between different ASRs – Deepspeech [28], Sphinx [29] and Google [30]. We found SFL and Causal explanations did well on all three ASR systems with respect to size and consistency.

Source code for `X-ASR` and examples from our experiment are available at `https://anonymous.4open.science/r/Xasr-6E11`.

4.2 Post-hoc Perturbation-Based Explainability Methods: A Unified Framework

The fundamental concepts and representative techniques of explainability were introduced earlier in Chapter 2. We also presented a unified framework for perturbation-based post-hoc methods in Chapter 2.3.2. That framework formalized the common steps shared by methods LIME, SFL, and Causal.

In this section, we will briefly revisit several representative explainability methods and the unified perturbation-based framework introduced in Chapter 2.

Explainable AI (XAI) techniques aim to increase trust in machine learning models by providing rationale for model decisions. However, the majority of XAI research has focused on classification tasks in image recognition and natural language processing fields. None of the existing techniques considers sequence outputs as seen with ASR transcriptions. The focus of this chapter is on post-hoc methods as these can be applied in settings with an existing prediction model. It is worth noting that the existing explanation methods can be divided broadly into either, perturbation-based or gradient-based. Perturbation-based methods perturb the inputs in the neighbourhood of a given instance to observe effects of perturbations on the model's output. Changes in the outputs are attributed to perturbed inputs and used to estimate their importance for a particular instance. Perturbation-based methods are black-box that do not consider model structure and can be easily applied to any model. On the other hand, gradient-based methods inspect DNN architecture and parameters to compute the contribution of all input features through a forward and backward pass through the network. Examples are Integrated Gradients in [60], DeepLIFT in [62], Grad-CAM in [61]. We do not consider gradient-based explanation techniques as they are white-box, requiring information on the DNN architecture and parameters. Many of the commercial ASR do not reveal their inner workings, so a white-box explanation technique is not easily applied. On the other hand, perturbation-based techniques being black-box are more generally applicable to any model, and we consider these in our work.

Perturbation-based methods share a common framework centered on modifying the input and observing how these changes affect the model's predictions. The general workflow includes four key steps:

- **Generate Perturbations:** Modify specific parts of the input to create variations.
- **Classify Mutants:** Test the model on these modified inputs and record how the predictions change.
- **Quantify Importance:** Analyze how each modified part impacts the output to determine its importance.
- **Construct Explanation:** Use the importance scores to rank input parts or find the smallest set of input parts needed to reproduce the output.

Among a lot of perturbation-based techniques, we selected three techniques: LIME [34], SFL [63] and Causal [36]. Their primary differences lie in how they handle the third step of quantifying importance. Below, we briefly describe their key characteristics in this step and the reasons for choosing them.

- LIME quantifies importance by approximating the model's local behavior using a simple, interpretable model, such as a linear model. The parameters of this surrogate model are treated as importance scores. LIME is widely used due to its simplicity and flexibility, often serving as a baseline method in explainability studies. Its broad adoption and straightforward design made it a natural starting point for our work.
- SFL starts to use statistical fault localization methods to compute importance. Instead of relying on a surrogate model, the SFL assigns importance scores based on statistical measures. This approach provides greater results in explanation size, accuracy, and speed, which inspired us to select it.
- Causal methods are grounded in causality theory [71], leveraging principles of cause-and-effect relationships to identify input components responsible for changes in the model's output. This theoretical foundation provides robust explanations, making it an essential and complementary approach in our work.

The technical details are shown in the background chapter 2.3.2.

4.3 Methodology

Applying explainability techniques such as LIME, SFL, and Causal to ASR models presents two major challenges:

- These methods are designed for tasks where models output a single label. However, ASR models produce sequences of varying lengths.
- SFL and Causal are developed for fixed-size image inputs, whereas audio inputs are variable in length.

To address these challenges, we propose `X-ASR`, a framework that generates explanations for ASR transcriptions through a two-step process:

1. *Classify:* We evaluate ASR transcriptions against a reference transcription using similarity metrics. This enables the use of perturbation-based methods by assigning binary correctness labels based on prediction changes.
2. *Adaption:* We extend SFL, Causal, and LIME to handle audio inputs by introducing speech-specific adaptations to variable-length audio, designing perturbation strategies that mask audio frames instead of pixels,

The following sections show the details.

4.3.1 Classifying ASR Transcriptions

Unlike in classification tasks, where it is relatively simple to determine whether an output label is Correct (matches the expected label) or Incorrect, assigning a binary label to ASR transcriptions is more complex. An ASR transcription may differ from the expected result but still be considered acceptable. For example, if the expected transcription is “I’d like an apple" and the ASR outputs “I like apple", we might still judge the transcription as correct. Evaluating the correctness of an ASR transcription often involves human judgement, taking into account small variations from the expected output.

To address this issue, we assign Correct or Incorrect labels to ASR transcriptions based on their similarity to the expected transcription, using a user-defined threshold, T. If the similarity to original transcription is higher than the threshold T, then the perturbed audio transcription is marked as Correct, and Incorrect otherwise.

Our framework, `X-ASR`, supports *two similarity metrics*:

1. `Bert` - We compute semantic vectors for the original and perturbed audio transcriptions using Sentence Bert [120]. We then report similarity between them as the Cosine similarity between the two vectors, similar to [120],[121] and [122].
2. `Word Error Rate(WER)` - We use `WER` to compare two transcriptions based on the number of insertions, deletions and substitutions compared to total words in original transcript ([100, 101]).

4.3.2 Adapting Explanations for ASR

We adapt three explainability techniques—SFL, Causal, and LIME—for ASR tasks by redefining their perturbation strategies to handle audio-specific input characteristics. SFL and Causal are all designed for image inputs, which are fixed-size 2D grids of pixels, audio signals are sequential and variable in length, requiring unique adaptations for input segmentation and mutation generation.

Adapting SFL Explanations The original SFL method creates mutants by masking chosen pixels in fixed-size 2D images, filling them with a neutral background (typically gray). Each pixel holds a single intensity value, making this straightforward in the image domain.

For ASR, we reinterpret pixels as audio frames, where each frame contains a vector of values rather than a single value. Masking a frame involves replacing all its feature values with zeros, effectively creating a silent frame in the audio.

Adapting Causal for ASR The original Causal explanation method is also designed for two-dimensional image inputs, where the input is partitioned into non-overlapping regions by selecting two random points along each dimension. These points define two perpendicular cuts, dividing the image into four disjoint regions. Each region is evaluated independently to determine its contribution to the model’s output.

For audio inputs, which have only one temporal dimension, we adapt this process by redefining the partitioning strategy:

1. We randomly select three distinct time points within the audio sequence, ensuring they are mutually different and within the valid time range.
2. These indices partition the audio into four disjoint regions: frames before the first index, frames between the first and second indices, frames between the second and third indices, and frames after the third index.
3. Each of these regions is treated as a separate evaluation unit, analogous to the regions formed in the original Causal.

The masking strategy for each region follows the same approach described in the SFL adaptation.

This adaptation maintains the original method's logic while accommodating the sequential structure of audio data.

Adapting LIME Explanations LIME is designed to just output the importance ranking of elements in our case, and this ranking is considered a LIME explanation. Inspired from minimal and sufficient explanations in SFL and Causal, we construct LIME explanations using a greedy approach that starts from the top ranked frame, and then adds frames in the rank list iteratively to the explanation until classification of the explanation is correct with respect to the original audio transcription. We use these potentially smaller LIME explanations in our experiments in Section 4.5

4.4 Experiments

We evaluate the explanations generated for ASR models using three quality metrics that are described below. We use three different ASR – Google API [30] (referenced as Google in the results), baseline model of Sphinx [29] and Deepspeech [28]) 0.7.3 version – and 1000 audio samples from CommonVoice dataset [90] and 1000 audios from Timit dataset [39]. Within `X-ASR`, we evaluate three explanation techniques mentioned in Section 4.3, namely, `SFL`, `Causal`, `LIME`, with two similarity metrics (`Bert` and `WER`) used to classify transcriptions from perturbed inputs. We use the default setting for every ASR.

We use the following parameters in our experiments: an audio sampling rate of $16000Hz$, frame length of 512. For SFL, Causal and LIME, the mutation factor is 0.05 which determines the proportion of frames to be randomly selected for each mutation. The size of the mutants set is 100. For Causal, we run the experiments 3 times to get a reliable ranking of frames. For similarity metrics, classification threshold for `Bert` is 0.5 and for `WER` is 0. We choose a zero threshold for `WER` to emulate strict classification. Additionally, we investigate the effect of choosing other classification threshold values for the similarity metrics and report our findings in Section 4.5.

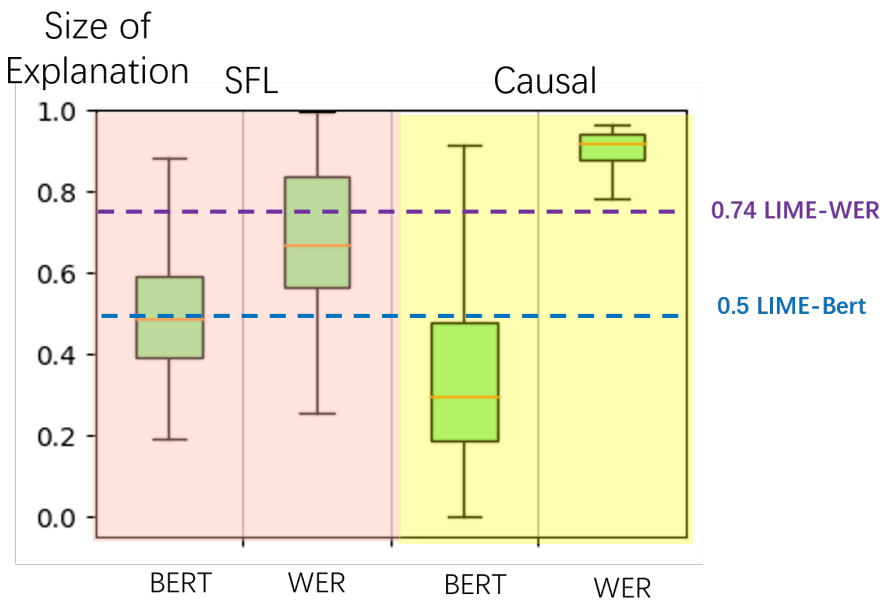
Quality Metrics for Explanations: We use two previously used quality metrics from image classification explanations [123, 35] to compare and contrast explanations over three ASR, namely, *size* and *consistency*.

Size We use size of the explanation as number of frames in the explanation versus the input audio. When size is smaller, the quality of the explanation is better as it indicates the technique is more selective in identify key frames contributing to the output. If a technique selects majority of the input audio frames, it is not very helpful in interpreting or understanding the rationale for model output. We compare size of explanations across different explanation techniques for every input audio sample.

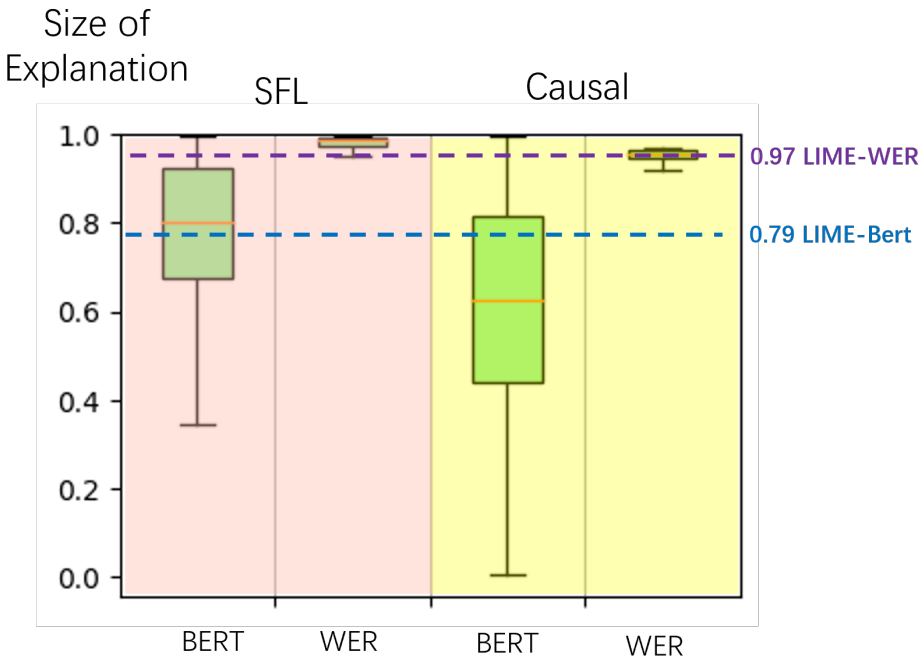
Consistency The consistency metric assesses the degree to which similar explanations are generated from different ASR for a given audio sample. Since transcriptions for an input audio are fairly consistent across ASR, we expect explanations to also remain consistent. However, explanations from different ASR may vary due to the variance of certain explanation methods.

We assess the consistency of explanation methods using Google ASR as the reference and calculate the fraction of frames that stay the same in explanations generated with other ASR, namely Sphinx and Deepspeech. Based on this definition, consistency is between 0 and 1, and a higher consistency in explanations across ASR is desirable.

4.5 Results and Analysis

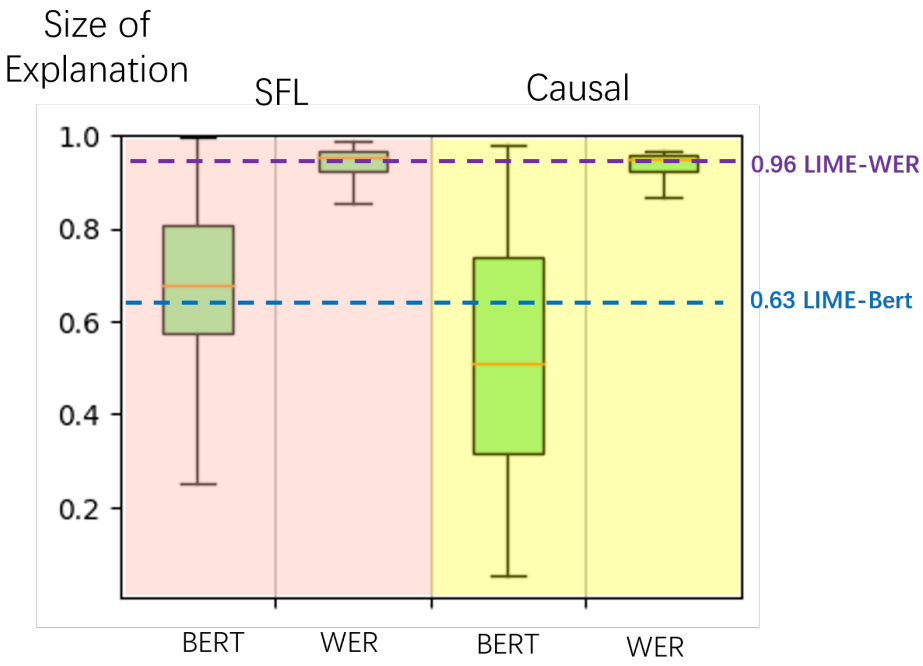


(a) Google



(b)

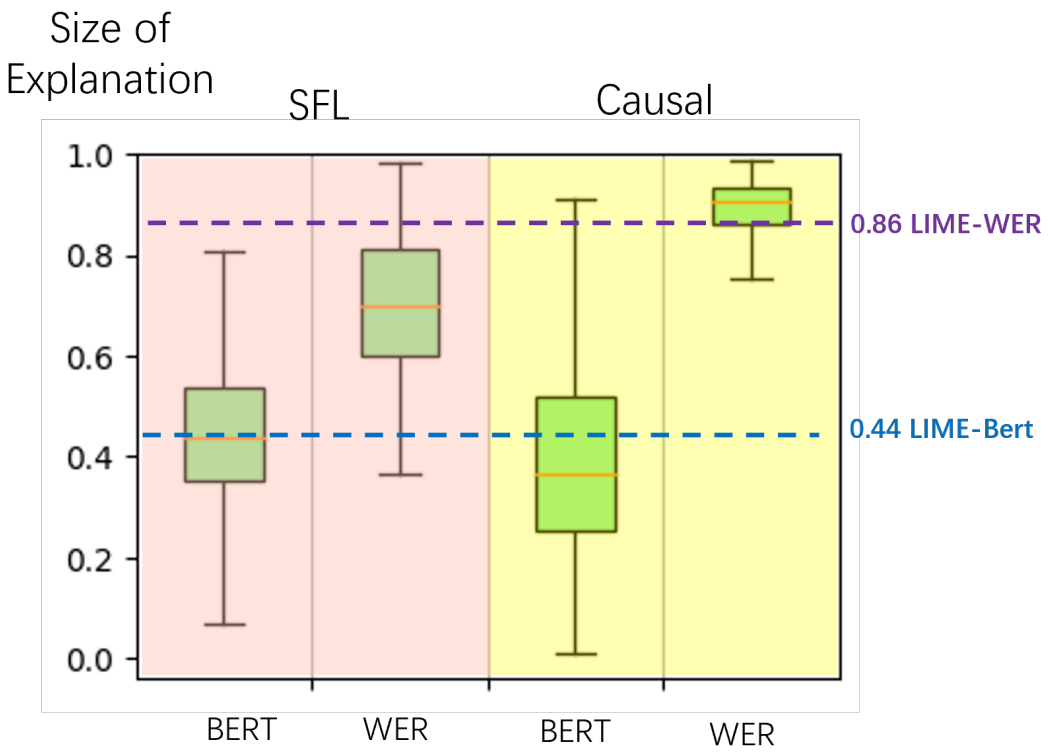
Sphinx



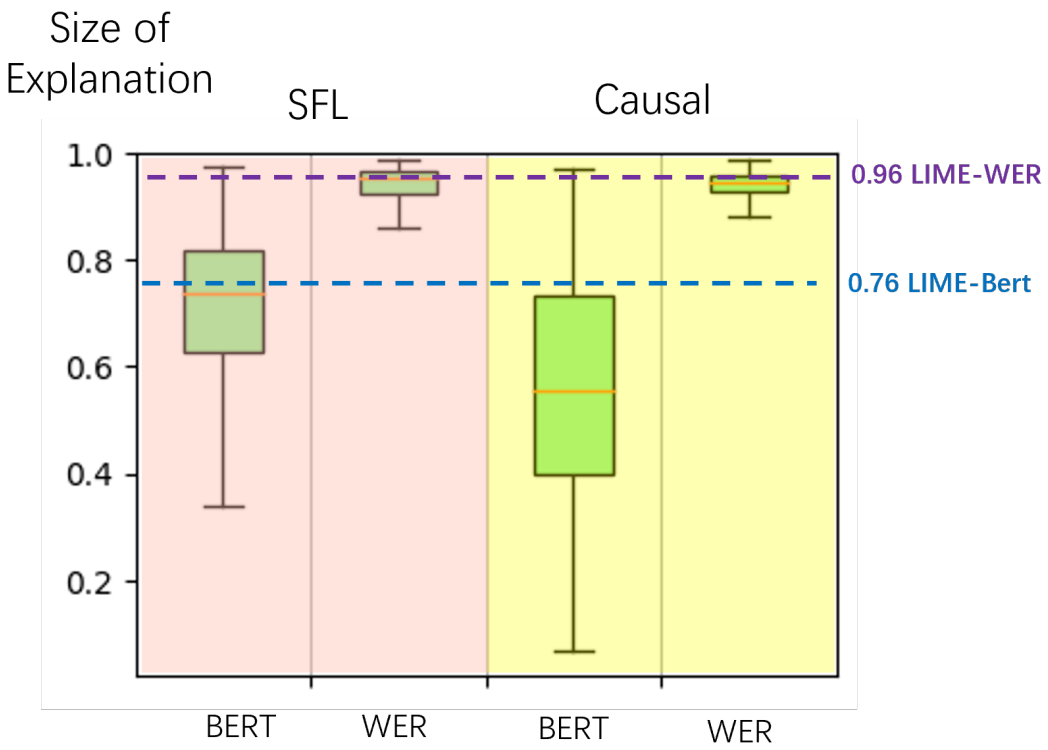
(c)

Deepspeech

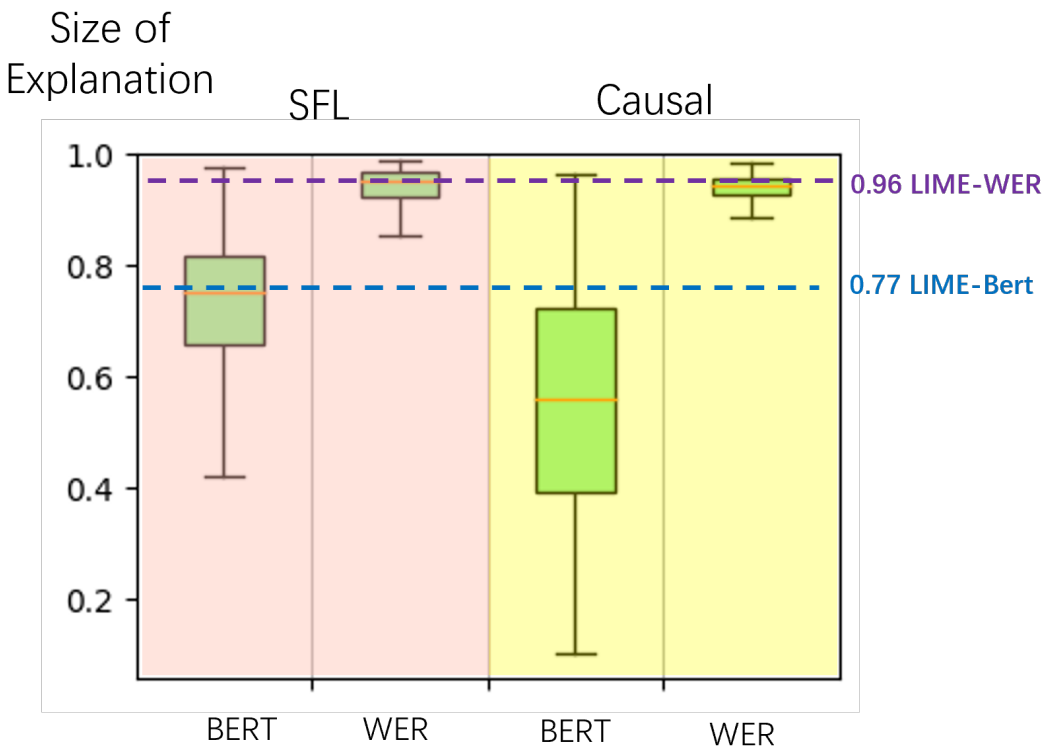
Figure 4.1: Size of explanations using SFL and Causal against LIME using each of two similarities on three different ASR systems, using 1000 samples from the CommonVoice dataset.



(a) Google



(b) Sphinx



(c) Deepspeech

Figure 4.2: Size of explanations using SFL and Causal against LIME using each of two similarities on three different ASR systems, using 1000 samples from the TIMIT dataset.

		SFL	Causal	LIME
Consistency (Google-Sphinx)	Bert	0.80	0.63	0.89
	WER	0.97	0.98	0.96
Consistency (Google-Deepspeech)	Bert	0.69	0.56	0.88
	WER	0.95	0.97	0.96

Table 4.1: Consistency(with respect to Sphinx or Deepspeech) of explanations generated by three explanation methods across two similarity metrics using Google ASR and 1000 samples from CommonVoice dataset.

		SFL	Causal	LIME
Consistency (Google-Sphinx)	Bert	0.76	0.61	0.84
	WER	0.97	0.98	0.97
Consistency (Google-Deepspeech)	Bert	0.78	0.62	0.82
	WER	0.97	0.98	0.97

Table 4.2: Consistency(with respect to Sphinx or Deepspeech) of explanations generated by three explanation methods across two similarity metrics using Google ASR and 1000 samples from Timit dataset.

	Threshold	SFL	Causal	Lime
Bert	0.25	0.50	0.30	0.50
	0.5	0.59	0.44	0.63
	0.75	0.66	0.58	0.73
WER	0	0.86	0.93	0.86
	0.5	0.70	0.68	0.66

Table 4.3: Size of explanations generated by three explanation methods across two similarity metrics using Google ASR and different thresholds.

	Threshold	SFL	Causal	Lime
Bert	0.25	0.68	0.24	0.57
	0.5	0.78	0.46	0.80
	0.75	0.94	0.78	0.94
WER	0	0.97	0.95	0.91
	0.5	0.90	0.76	0.86

Table 4.4: Consistency(with respect to Sphinx) of explanations generated by three explanation methods across two similarity metrics using Google ASR and different thresholds.

All three techniques in our method, `SFL, Causal, LIME`, successfully generated explanations for every input audio across the CommonVoice and TIMIT datasets and for all three ASR models. This section presents a detailed evaluation of `SFL, Causal, LIME`, focusing on size and consistency of explanations. Results are analyzed separately for the CommonVoice and TIMIT datasets, with an additional look at the influence of threshold settings.

4.5.1 Size of Explanations

Figures 4.1 and 4.2 illustrate the size of explanations generated by the different techniques across three ASR models on 1000 samples each from the CommonVoice and TIMIT datasets, respectively.

Across all datasets and ASR models, we find that Causal outperforms LIME with the `Bert` similarity metric by producing significantly smaller explanations (a statistically significant difference confirmed via one-way ANOVA followed by post-hoc Tukey's test [124]). This is due to Causal's approach, which not only accounts for output changes after masking a region but also considers the number of other regions affected by masking. Additionally, Causal generates smaller explanations than SFL (statistically significant) due to its stricter causal-theory conditions in assigning region responsibility.

Additionally, when we fix the similarity metric and explanation method, Google ASR consistently generates the smallest explanations compared to other ASR models. For example, on the TIMIT dataset with the Bert similarity metric, LIME generates explanations with a size of only 0.44 on Google ASR, significantly smaller than the sizes of 0.76 on both Sphinx and Deepspeech. This outcome suggests that Google ASR is able to derive accurate transcriptions from fewer frames compared to Sphinx and Deepspeech.

Finally, there is a difference in explanation size based on the similarity metric. WER generates larger explanations than Bert due to WER's strict intolerance for transcription differences, which results in more perturbations satisfying the condition for output classification changes. Bert, with a threshold of 0.5, allows for minor transcription variations, producing smaller explanations. Raising the threshold for WER similarly yields more concise explanations. Experiments with other thresholds in both metrics followed a similar trend, with results presented later.

4.5.2 Consistency

Table 4.1 and Table 4.2 show the consistency of explanations between Google ASR and Sphinx (rows highlighted in pink) and Google and Deepspeech (rows highlighted in yellow) on samples of 1000 from CommonVoice, and 1000 from TIMIT, respectively.

In most cases, we find that LIME and SFL achieve similar consistency, with both being more consistent than Causal under the Bert similarity metric on both ASR pairs. For example, on the TIMIT dataset in Table 4.2, for the Google-Deepspeech comparison using Bert, LIME achieves a consistency of 0.84, while SFL reaches 0.78—both considerably higher than Causal's 0.61. Causal, on the other hand, is most consistent with the WER similarity metric, which can be attributed to its consistently larger explanation size. The same pattern is observed for LIME explanations when they are larger in size.

These results demonstrate that size and consistency are often conflicting metrics. We find that SFL with the Bert metric offers a good balance between the two, providing both concise and reasonably consistent explanations.

4.5.3 Impact of Different Threshold

To explore the impact of different thresholds, we randomly select 20 samples from CommonVoice dataset due to the time-consuming nature of the experiments and generate their explanations at varying thresholds. We then evaluate these explanations using the two metrics outlined in the paper: size and consistency.

For Bert, we test thresholds of 0.25, 0.5, and 0.75. With Bert, A higher threshold demands greater similarity between transcriptions to classify them as Correct, making the classification stricter. In contrast, for WER, we explored thresholds of 0 and 0.5. A higher threshold in this case relaxes the similarity requirement, allowing for more lenient classifications.

Table 4.3 shows the size of explanations generated by the different techniques across similarity metrics under various threshold settings. We find that across similarity metrics, a lower similarity requirement for correct classification results in smaller explanation sizes. For Bert, a threshold of 0.25 produces the smallest explanations, while for WER, this threshold is 0.5.

It is worth noting that changes in threshold do not affect the relative performance of the explanation techniques. For instance, Causal consistently generates smaller explanations than SFL and LIME across all tested Bert thresholds.

Table 4.4 highlights the consistency of explanations between Google ASR and Sphinx under these thresholds. Stricter thresholds, such as `Bert` at 0.75, improve consistency across systems, aligning with the larger explanation sizes observed in Table 4.3. This suggests that higher thresholds enhance stability.

In summary, stricter thresholds produce larger and more consistent explanations, while relaxed thresholds reduce explanation size at the cost of stability.

4.6 Conclusion

This chapter examined the application of perturbation-based post-hoc explainability methods to ASR, addressing **RQ2: How effective are post-hoc explainability methods when applied to speech AI, particularly ASR, and how can their reliability be systematically evaluated?** We proposed the first framework, `X-ASR`, which supports three representative techniques: SFL, Causal attribution, and LIME. The results showed that SFL and Causal produced more compact and consistent explanations than LIME, while all three techniques achieved low redundancy. Stability and consistency metrics indicated that explanation quality was sensitive to the choice of similarity measure, with WER-based similarity favouring larger but more stable explanations. These findings represent one of the first systematic analyses of post-hoc explainability in the speech domain.

Several limitations should be noted. First, perturbation-based attribution is computationally expensive, as each explanation requires multiple forward passes through the ASR model. Second, explanation stability is sensitive to perturbation granularity and metric choice, raising concerns about reproducibility. Third, the study focused only on portable perturbation-based methods without leveraging ASR model internals, which restricts insights into model-specific mechanisms. Finally, while these methods identify influential acoustic segments, they provide limited information about higher-level linguistic or prosodic features, constraining their interpretability for end users.

Despite these limitations, the results provide a foundation for systematically evaluating explainability in ASR.

Chapter 5

Explainability: Phoneme-Level Validation of ASR Explanations

Building on the previous chapter, where we applied post-hoc explainability techniques to black-box ASR systems, we observed that while these methods can provide intuitive interpretations, they often lack objective validation. In particular, it remains unclear whether the highlighted segments identified by explanation methods truly correspond to phonetic regions critical for recognition. This chapter addresses this gap by introducing a phoneme-level validation framework to assess the reliability of explanations. The chapter is structured as follows: we begin with an introduction, followed by methodology, experimental setup, results and analysis, and conclude with a summary of key findings.

The novelty of this chapter lies in proposing a structured evaluation methodology that connects post-hoc explanations to phoneme recognition performance, thereby enabling quantitative assessment of explanation faithfulness. By grounding evaluation in speech-specific tasks, this chapter establishes one of the first systematic approaches to benchmarking XAI in ASR.

5.1 Introduction

Although current explanation techniques have significantly advanced our understanding of DL model predictions, the reliability of these explanations has been largely overlooked. Some recent studies [125, 126] have demonstrated the limitations of current XAI techniques. For instance, [125] applied three different XAI techniques on a CNN-based breast cancer classification model and found the techniques disagreed on the input features used for the predicted output and in some cases picked background regions that did not include the breast or the tumour as explanations.

Literature on evaluating the reliability of XAI techniques is still in its nascency and can be broadly divided into two branches - (1) Studies that assume the availability of expert annotated ground truth, maybe in the form of bounding boxes for images, to evaluate the accuracy of explanations [127, 128, 129, 130] and (2) research that uses the idea of removing relevant

(or important) features detected by an XAI method and verifying the accuracy degradation of the retrained models [131, 132, 133, 134, 135]. The first category requires human-annotated ground truth for evaluation while the second category incurs very high computational cost to verify accuracy degradation from retraining the models.

Research on explanations for ASR is still in its early stages. We modified image-based explanations for speech input in ASR, but the validity of the explanations was not evaluated. A key challenge in ASR is the absence of a clear mapping from words output to segments of audio, owing to the fact that ASR outputs are generally influenced by surrounding words, not just the immediate speech input.

Why Timit PR task?

In an effort to evaluate the reliability and trustworthiness of explanations in the ASR context, we use the TIMIT [39] Phoneme Recognition (PR) task using the standard recipe from the Kaldi toolkit [38], as a **simple but basic controllable** task that is predictable with a phoneme language model with ground truth in the form of manual labeling and segmentation at the phomeme level. At this early stage, we believe it is important to properly validate the technique, and such detailed ground truth labelling – not found in any other data set – is essential for evaluating both the quality and reliability of the explanations generated.

We generate explanations for the PR system via LIME [34], a renowned XAI method designed for images. It employs a linear regression model to locally approximate the black box DL model's prediction. We chose LIME due to its local perturbation approach, aligning with our belief that PR models exhibit local effects: phonemes are primarily influenced by adjacent phonemes, not distant ones.

To adapt LIME to produce explanations for the TIMIT PR task, first, we classify every phoneme in a PR transcription as correct or incorrect based on comparison with the expected transcription. Second, we apply LIME to generate explanations for each phoneme in the transcription using input speech perturbations. Third, we improve performance of LIME by focusing perturbations of the input audio to be within a limited window around the phoneme of interest using two LIME variations, LIME Window Segment(LIME-WS) and LIME Time Segment (LIME-TS). A segment refers to a distinct section within an audio and each of LIME-WS and LIME-TS has its own definition of segment which is then used as the basic unit of an explanation. We evaluate reliability of the basic LIME explanations and the variants, LIME-WS and LIME-TS, for the TIMIT PR task on Kaldi using the ground truth labelling of the TIMIT dataset. We found explanations with LIME-TS are the most reliable for the TIMIT Phoneme Recognition task using Kaldi, capturing the ground truth 96% of the time in the top three segments of the explanation. LIME-TS outperforms LIME and LIME-WS by up to 44.8% and 17% on All speakers, respectively. Source code for `X-PR` and examples are available at `https://anonymous.4open.science/r/X-PR-4560`.

5.2 Methodology

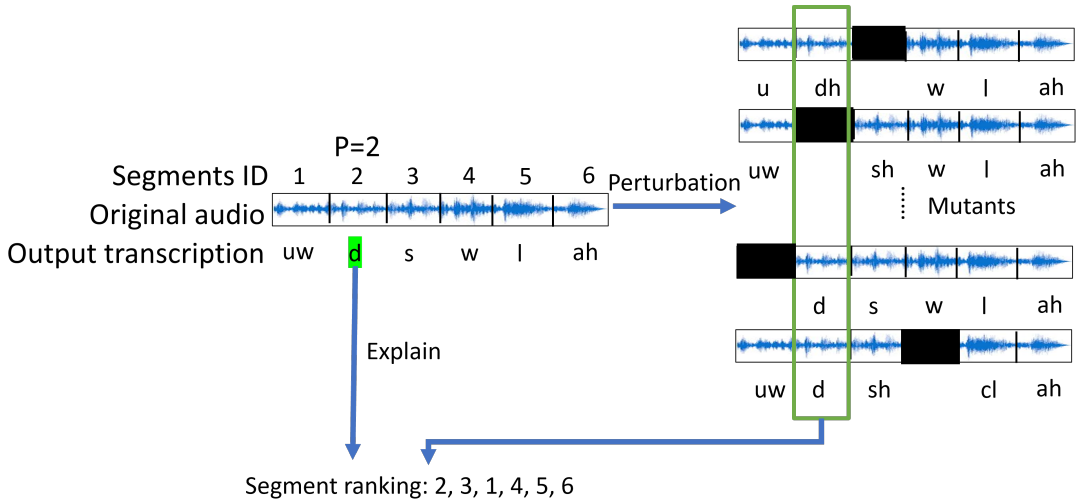


Figure 5.1: An outline of generating an explanation for a phoneme appearing in the output transcription.

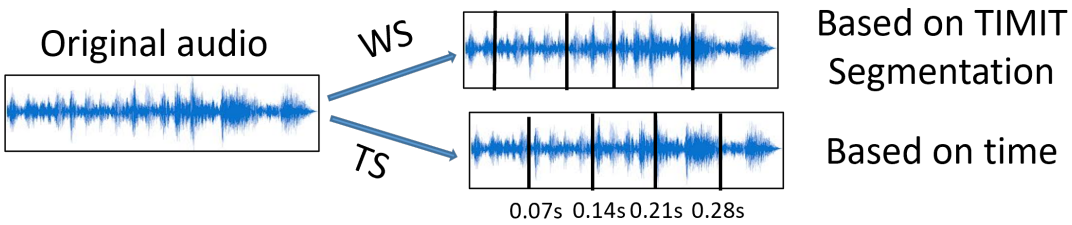


Figure 5.2: Different segmentation used by (LIME-WS,LIME) and LIME-TS.

Motivation

Phoneme recognition systems, especially those based on complex algorithms, often function as a black box, making their decision-making opaque. For example, if a system misidentifies the phonemes ‘b’ and ‘p’ in words like ‘bat’ and ‘pat’, the developers need to understand why. Explanation techniques are designed to clarify such confusions, offering insight into the system’s decisions. This not only helps in refining the system but also builds developer and user trust.

What is an Explanation for a Phoneme?

Taking cues from existing explanation methods helps us understand how to design explanations in other domains. In image tasks, models often highlight the areas of eyes, to explain why this image is labeled as a face. Similarly, for PR, an explanation should pinpointing key frames of speech for recognizing a phoneme. For example, as shown in the Figure 5.1, we have an input audio (divided into segments available in TIMIT) , alongside its original transcription. For the phoneme ‘d’ which appears in the transcription, the explanation

is the importance ranking of segments in the audio. The higher the rank, the more crucial that segment is for recognizing ‘d’. As shown in the Figure 5.1, the second segment emerges as the most important or the highest ranked for the phoneme ‘d’.

A general Framework of generating an Explanation:
Figure 5.1 presents a high level overview of generating an explanation for a phoneme in the transcription output. We start with an input audio and its segments. We then perturb the audio by masking out segments randomly. To clarify, masking is done by setting the sample points of chosen segments to zero, creating silent intervals in the audio. In Figure 5.1, segments shaded in black within each mutant are the ones that have been masked. For the phoneme of interest, ‘d’ in Figure 5.1, we compute the importance ranking of segments in the audio as an explanation for ‘d’ which is based on the effect of the perturbations on the phoneme output. It is worth noting that when the segment corresponding to phoneme ‘s’ is masked (see the first mutant), we find the adjacent phoneme ‘d’ is wrongly recognized as ‘dh’. This is because the phoneme ‘d’ is affected when neighboring segments are masked.
We delve deeper into the LIME explanation approach and its variants, LIME-WS and LIME-TS, later in this section. To apply the LIME technique, we first need to treat the PR task as a classification task. To do this, we attach 0 or 1 label to every phoneme in the output transcription by aligning and comparing it with the expected transcription that is available in the TIMIT dataset. We implement classification of each phoneme output with the NIST sclite scoring tool[1].

5.2.1 Explanations using LIME and its variants for PR

In this section, we start by describing the base case which is a straight forward adaptation of LIME to work on the PR task. We then describe our variants, LIME-WS and LIME-TS, that applies perturbations to segments within a fixed window.

Base LIME Explanations:
LIME, proposed in [34], is a black-box XAI technique that can be applied to any model without requiring information on its structure. Given a complex neural network (NN) model $f(x)$ that takes in an input x and produces an output y, the goal of LIME is to mimic the behavior of a complex model $f(x)$ with a weighted linear regression model $g(x)$ in the local area of a specific instance of interest x. The weighted linear regression model $g(x)$ is defined as $g(x) = w_0 + w_1x_1 + w_2x_2 + ... + w_dx_d$, where w_0 is the intercept term, w_1 to w_d are the weights assigned to each feature, and x_1 to x_d are the feature values of the instance x. Among them, w_1 to w_d denote the *importance score* of each feature. For the PR task, the equation is defined as:

$$g^P(x) = w_0 + w_1x_1 + w_2x_2 + ... + w_dx_d \tag{5.1}$$

1. https://github.com/usnistgov/SCTK

The complex NN model $f(x)$ is the Kaldi system. P refers the positional index of the phoneme to be explained in the original output. As shown in the Figure 5.1, the specific instance x is the input audio, 'd' is the phoneme output to be explained and P is 2. To fit $g^P(x)$ and get w_1 to w_d, LIME needs several perturbed instances of x and their outputs. As shown in the right side of the Figure 5.1, mutants of the original audio x are created by masking out segments randomly. Input features x are the segments. Values of features x_1 to x_d in the mutant are 1 or 0, where 1 means that the segment at this position has not been masked while 0 implies the segment is masked. For example, in the Figure 5.1, for the uppermost mutant, the third segment has been masked. Consequently, the value of x_3 is set to 0, with all other feature values being 1.

For each mutant m_i, we align its transcription against the original (unmasked) output transcription, y. All transcriptions are from the trained Kaldi. After alignment, we may find some phonemes match the original output transcription while some others are incorrect. For those correct phonemes, the output of m_i - represented as $f^j(m_i)$ - will be 1 while for others, $f^j(m_i)$ will be 0. j refers the index of the positional index of phonemes in the original transcription. Continuing with the example of the uppermost mutant, denoted as m_1, we observe the following: 'd' to be explained (at the second position) has been wrongly identified as 'dh'. Therefore, we have $f^2(m_1) = 0$, which is also the $f^P(m_1)$. Similarly, the fourth element, 'w', remains unaltered, resulting in $f^4(m_1) = 1$. Then, the LIME model will compute how the masked segment in each of the mutants affects the output phoneme 'd' (bounded with a green box in the Figure 5.1). If the masked segment changes the output phoneme 'd' at that position, then it will have a high ranking (aggregated over many mutants with masked segments).

Using the mutants and the associated binary labels $f^P(m_i)$ after alignment to original transcription, LIME will start to fit $g^P(x)$ using the locally weighted least squares objective function, which is defined as:

$\text{L(g)}_P = \sum_{i=1}^{n} \varepsilon_i (f^P(m_i) - g^P(m_i))^2$(5.2) In this equation, n is the number of mutants and ε_i is a weight assigned to each mutant m_i that reflects its closeness to the audio of interest x. It is computed as the cosine similarity between the instance x and the mutant m_i, which is $\varepsilon_i = Cosine\_Similarity(x, m_i)$. The weights w_1 to w_d in the fitted linear regression model, $g^P(x)$, indicate the importance score of different audio segments for the selected output phoneme. We treat the ranking of segments based on their importance score as the explanation for each phoneme, as shown in Figure 5.1.

Segment-based LIME with a sliding window (LIME-WS):
In Base LIME, mutants for the original audio x are created by masking random segments. However, given the local nature of PR task, distant audio segments likely will not impact the phoneme in focus. Considering this, removing ineffective mutants can optimize computation.
To realize this idea, we use a fixed length sliding window during the generation of perturbations for LIME explanations. The sliding window slides from left to right one segment at a time. Within the range delimited by the sliding window, a pre-determined number of segments are randomly chosen for masking, while keeping the segments outside the sliding window unchanged. We hypothesize that focusing on perturbations within this sliding window will result in higher quality explanations. Other steps in LIME-WS for fitting the linear regression model remain the same as LIME.

Time Segment-based LIME with a sliding window (LIME-TS):
LIME and LIME-WS employ audio segmentation from the TIMIT dataset, segmented by linguistic experts. However, manual segmentations might be absent in common datasets like Librispeech and Common voice[90]. To overcome this challenge and generalize the applicability of our segment-based explanation technique, we investigate a method to split audio into uniform, non-overlapping segments via timestamps. Figure 5.2 contrasts the audio segmented from TIMIT and by timestamps. We split the audio into $70ms$ segments (to remain comparable with average length of manual segments), but this can be changed based on user choice.

5.3 Experiments

We evaluate the reliability of explanation techniques using the TIMIT PR model from Kaldi. For generating explanations, we choose the TIMIT dataset owing to the ground truth mapping of phonemes to input speech and details like speaker's gender. We generated explanations for all 630 speakers using their shared 'SA1' sentence. This uniform sentence allows comparison of explanation techniques across different demographics, focusing on the impact of factors like gender.

5.3.1 Validity Metric

The validity metric evaluates the reliability of the three explanation methods – LIME, LIME-WS, and LIME-TS. The metrics are defined as $validity_1 = \frac{N_1}{N}$, $validity_3 = \frac{N_3}{N}$, and $validity_5 = \frac{N_5}{N}$, where N is the total number of phonemes, N_1 is phonemes with the top-ranked segment matching the ground truth, while N_3 and N_5 represent phonemes where the ground truth is within the top 3 and 5 ranks, respectively. For all three metrics, higher is better.

5.4 Results and Analysis

	LIME / Random ranking			LIME-WS / Random Ranking			LIME-TS / Random Ranking		
	All	Female	Male	All	Female	Male	All	Female	Male
$validity_1$	0.40/0.0	0.35/0.0	0.42/0.0	0.49/0.03	0.43/0.03	0.50/0.03	0.86/0.03	0.84/0.02	0.86/0.03
$validity_3$	0.62/0.06	0.54/0.06	0.62/0.06	0.76/0.09	0.72/0.09	0.77/0.09	0.96/0.12	0.94/0.11	0.96/0.13
$validity_5$	0.67/0.13	0.59/0.12	0.67/0.13	0.83/0.15	0.81/0.14	0.84/0.15	0.97/0.16	0.96/0.15	0.97/0.16

Table 5.1: $validity_1$, $validity_3$ and $validity_5$ of explanations generated by three explanation methods and randomly sorted method (Right side of every slash) across gender using Kaldi PR.

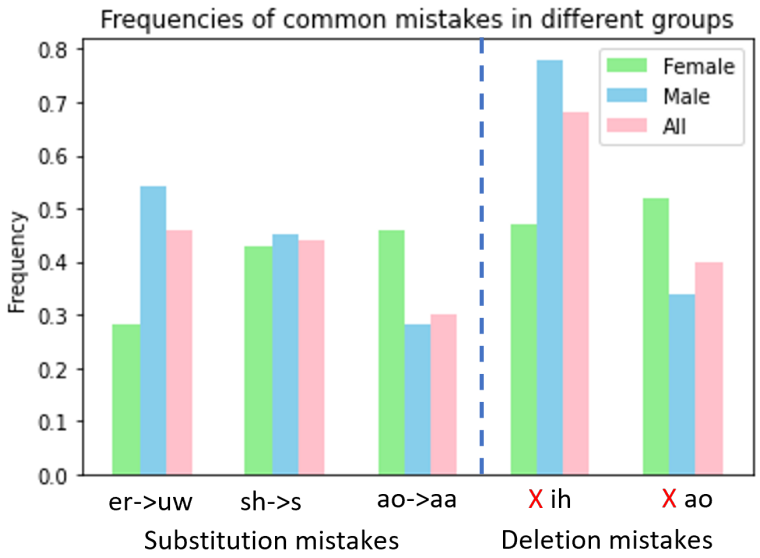


Figure 5.3: The top five most frequently occurring transcription mistakes and their corresponding frequencies on different groups. There are three substitution mistakes on the left of the dashed blue line and two deletion mistakes on the right. For example, $er \rightarrow uw$ means that er is replaced by uw and Xih means that ih is deleted.

5.4.1 Comparison of Explanation Techniques.

The three methods, LIME, LIME-WS, and LIME-TS, successfully generate explanations for all audio samples in our dataset. In Table 5.1, we display the metrics $validity_1$, $validity_3$, and $validity_5$ for each technique across All speakers, and then segregated for Male and Female speakers. To benchmark these results, we also present validity scores obtained from random rankings—these values are shown after the $/$ symbol in the table. It is evident that all techniques are more trustworthy than random rankings across all metrics. For example, in Table 5.1, LIME-WS scores 0.49, 0.76, and 0.83 across the validity metrics for All speakers, greatly surpassing the scores of 0.03, 0.10, and 0.15 from random ranking. Both LIME and LIME-TS also exceed the random ranking significantly, with the differences verified statistically using one-way Anova followed by post-hoc Tukey's test [124]).

All	Female	Male
Segment Position (Phoneme output)	Segment Position (Phoneme output)	Segment Position (Phoneme output)
8 (d)	8 (d)	34 (er)
9 (aa)	9 (aa)	8 (d)
34 (er)	7 (dcl)	32 (y)

Table 5.2: The top three important segments in LIME-WS explanation for the $er \rightarrow uw$ mistake and the corresponding phoneme outputs in paranthesis with speaker groups All, Female and Male.

From Table 5.1, we see that LIME-WS and LIME-TS consistently perform better than LIME across all metrics for each speaker group. For example, on $validity_5$ with All speakers, they surpass LIME by 24% and 44.8% respectively. This outcome is in line with our expectations, as the sliding window introduced in LIME-WS and LIME-TS ensures that perturbations are restricted to a local range. This restriction enables the explanation technique to focus on relevant segments as the influence of a phoneme is typically confined to a small number of adjacent phonemes.

For LIME-TS versus LIME-WS, we find LIME-TS performs better than LIME-WS in all cases. We verified this difference is statistically significant using one-way Anova and a follow-up Tukey's HSD test [124] at a 5% significance level. For example, LIME-TS outperforms LIME-WS by 17% on $validity_5$ over All speakers. LIME-TS utilizes fixed-length time segments as the fundamental unit for generating explanations and additionally, LIME-TS segments are slightly smaller than LIME-WS segments – 70 versus 78 ms. The corresponding ground truth explanation in the original audio always overlaps with multiple LIME-TS segments(usually 2) which are considered equally significant. Conversely, LIME-WS uses manually labeled audio segments as the unit of explanation that has a one to one correspondence with the ground truth explanation. We believe the smaller LIME-TS segments for perturbation and the validity measurement (top 1, 2 or 5 ranked segments) that considers ground truth overlapping segments as equally important helps LIME-TS look more attractive than LIME-WS.

Overall, we find LIME-TS to be most reliable among the three explanation methods, capturing the ground truth in 96% of the cases when considering the top 3 segments in the explanations. Additionally, it is easily generalizable, owing to its use of time-based segments rather than an expert labelled audio segment, as in LIME or LIME-WS.

5.4.2 Male versus Female Speakers.

Table 5.1 shows that the validity scores for all three explanation methods are higher for Males compared to Females. This is consistent with the fact that the TIMIT dataset, used for training, contains 70% male speakers and only 30% females. Thus, Kaldi better recognizes the nuances of male speech, even when both male and female speakers are uttering the same sentence.

We explored the most commonly occurring transcription errors in the different speaker groups. Figure 5.3 illustrates the five most commonly occurring transcription errors and their corresponding frequencies across all three speaker groups. Except for the $sh \rightarrow s$ substitution, we observed significant variances in error frequencies between Female and Male speakers. For example, the $er \rightarrow uw$ error is more prevalent among Male speakers (0.54) than Female speakers (0.28). Using a Wilcoxon Signed Rank Test with a 5% significance level, we verified that these error frequency differences between Males and Females are statistically significant. Explanations can help investigate possible causes for difference in error frequencies between genders. For instance, we examine the $er \rightarrow uw$ error. When comparing the LIME-WS explanations for Male and Female speakers regarding this error, we notice distinct patterns. Table 6.2 shows the three most recurring segments for this error. For Female speakers, these segments frequently center around position 8. In contrast, Male speakers often have two of their top three segments near position 34. Further analysis reveals the $er \rightarrow uw$ error appears at two key locations in sentence SA1: position 6 (surrounded by phonemes 'vcl' 'd' 'aa') and position 34 (adjacent to phoneme 'y'). Among Female speakers, 64% of these errors align with position 6, compared to 47% for Male speakers. This insight suggests potential areas for model improvement, tailored to each gender. Overall, explanations serve as a valuable tool to examine errors and compare speaker groups.

5.5 Conclusion

This chapter introduced a quantitative evaluation of post-hoc explainability in ASR by grounding explanations in the Phoneme Recognition (PR) task, thereby further addressing **RQ2: How effective are post-hoc explainability methods when applied to speech AI, particularly ASR, and how can their reliability be systematically evaluated?** Using the availability of ground truth phoneme alignments in TIMIT, we adapted LIME to speech and proposed two variants, LIME-WS and LIME-TS. Controlled experiments showed that LIME-TS was the most reliable, with the ground truth audio segment included in its phoneme-level explanations in 96% of cases. These results demonstrate that PR provides a suitable testbed for assessing explanation quality in a quantitative and interpretable manner.

Nevertheless, this evaluation also has limitations. It is restricted to the PR task, which benefits from frame-level ground truth alignments that are not available in more complex ASR tasks involving language models and long-span dependencies. As such, while PR offers valuable insights into the reliability of explanation methods, further work is needed to extend these evaluations to end-to-end ASR systems and more realistic speech applications.

Taken together, the results of this chapter provide the first systematic quantitative evidence of explanation reliability in ASR, reinforcing the broader conclusions of Chapters 4 and 5 and establishing a foundation for future benchmarking of XAI in speech.

Chapter 6

Intrisic Explainable SV System

The previous two chapters 4 and 5 demonstrated that post-hoc explanation methods can help identify input regions associated with ASR outputs. However, such methods remain fundamentally limited: they operate externally to the model, offering insights into input-output associations without revealing how decisions are formed internally. To address this limitation, this chapter turns to intrinsic explainability—an approach that embeds interpretability into the model architecture itself.

We select speaker verification (SV) as the target task for investigating intrinsic explainability because its modeling structure is naturally suited to concept-based interpretation. Unlike ASR, which requires detailed frame-level alignment between input and output, SV systems summarize entire utterances into fixed-length embeddings that reflect high-level speaker characteristics. This global representation provides a more flexible setting for aligning internal model dimensions with human-understandable attributes.

6.1 Introduction

Acquiring speaker-discriminative features is a critical step in various speaker recognition tasks, including speaker verification(SV). An effective SV system should effectively capture the differences between speakers. The aspects of speech that define a speaker's identity are diverse, encompassing both the physical characteristics of the vocal apparatus (such as gender, age, and certain medical conditions) and linguistic features like accent, dialect, native language, and sociolect (which can be influenced by education level and profession, affecting aspects such as lexicon, syntax, and stylistics). When humans listen to unfamiliar speech, they instinctively infer many of these attributes to form a mental image of the speaker. This approach is commonly applied in forensic phonetics [136, 137] and acoustics, where experts classify speakers in criminal investigations. Therefore, it is reasonable to expect that an effective SV should capture the various attributes that contribute to our understanding of speaker identity. Prior research has confirmed the significance of these attributes by demonstrating that the complex deep speaker embeddings produced by the traditional SV encode a broad range of speaker-related information and meta-information, such as emotion [138, 139], accent, language, gender, channel, and transcription details [140, 141].

However, the specific impacts of these attributes in the SV models are not well understood or explored, showing the need for a more detailed examination of their roles. Meanwhile, a challenge remains: traditional SV models often hide these attributes within their complex network structures, making it difficult to directly see their influence on performance.

To address this issue, we propose a study that explicitly examines the impact of group-level speaker attributes on SV performance in an explainable way. Our approach involves a two-stage model. In the first stage, classifiers are trained to identify attributes. In the second stage, these identified attributes are then incorporated into the SV model to get the final classification. This method aims to determine whether and how much these attributes affect SV system outcomes. Unlike traditional SV methods that mix these attributes into complex representations, **our model explicitly integrates these features from the ground up, using them exclusively to generate the final outcomes in a fully transparent way. This approach will also offer the importance of each attribute, ensuring an interpretable process that aligns with human understanding.**

The use of group-level attributes in SV is not entirely new. In 2020, Luu et al.[142] aimed to boost performance by jointly training deep speaker embeddings with attributes like age and gender, but they did not incorporate these attributes directly into the SV process. Our method, by contrast, directly utilizes personal attributes to make SV decisions. We achieve this through a Concept-Bottleneck Model (CBM) [76], which introduces an intermediate bottleneck layer for attribute classification in the first stage, and then uses these predictions for speaker classification in the second stage.

We adopt the CBM architecture in this chapter because it provides an intrinsic form of explainability by explicitly aligning intermediate representations with human-understandable attributes. Unlike post-hoc methods, which only analyse model behaviour after training, CBMs integrate interpretability directly into the model design: predictions are mediated through a set of concepts that can be inspected and evaluated. This makes CBMs particularly well-suited for our research aims, as they allow us to explore whether attributes such as accent, gender, and profession can serve as interpretable factors in speaker verification. By constraining the model to use these intermediate concepts, CBMs enable explanations that are both transparent and tied to measurable properties of speech, offering a complementary perspective to the post-hoc approaches investigated in earlier chapters.

It is worth noting that our aim is not to produce the best possible SV performance but rather to explore what can be achieved using only speaker's group-level attributes, within an explainable framework. Evaluated on the Voxceleb1 test set, our system demonstrates performance comparable to the ground truth when all correct attributes are used, proving its efficacy. We find to our surprise that our initial expectations of poor performance when relying solely on these attributes was entirely invalidated; our findings revealed that incorporating the group level attributes solely is effective. Source code available at `https://anonymous.4open.science/r/explainable-SV-E3C2`.

6.2 Explainable Attribute-Based SV

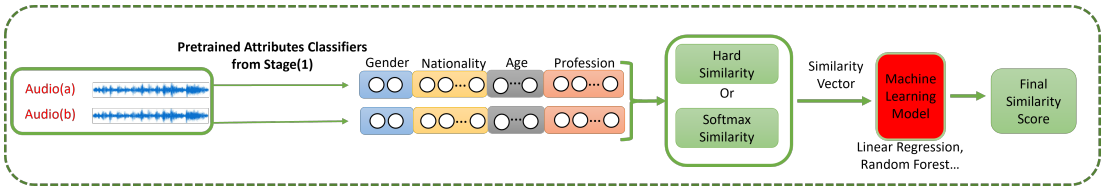


Figure 6.1: `Stage-2` of our SV system is shown. Pretrained classifiers from `stage-1` is used to extract attribute labels from pairs of audios. These attributes are then fed into a computation block that calculates a similarity vector for this pair of audio using hard or softmax similarity. The similarity vector is then used to train a `stage-2` Machine Learning Model, shown in red, which is the only component being trained during this stage. The output is the final similarity score, showing the likelihood that the two audio inputs are from the same speaker.

To develop an explainable speaker verification (SV) system that explores the impact of speaker attributes, we adapt the Concept-Bottleneck Model (CBM) [76], which introduces an intermediate layer to explicitly learn and represent human-understandable concepts. A predictor then uses these concept predictions to determine the final label. CBMs have been successfully applied in medical imaging, as Koh et al. [76] demonstrated, where CBMs identify critical indicators like lung opacity from X-rays before making diagnoses.

Similarly, our SV system uses a two-stage pipeline. In the first stage, attribute classifiers (`stage-1`) predict attributes such as gender, age and nationality from audio features. In the second stage (Figure 6.1), a `stage-2` model uses these attributes to perform speaker verification. We explain both stages in detail below.

6.2.1 Stage-1: Attribute Classifiers

In the first stage, we train multiple classifiers, each targeting a specific attribute, using two approaches to optimize attribute prediction.

Xvector and ECAPA: In the first approach, we use speaker embeddings from two pre-trained SV models—Xvector [53] and ECAPA [54]—as inputs for the `stage-1` classifiers. These embeddings enable our classifiers to capture complex speaker characteristics by leveraging the rich information provided by the pretrained models. Throughout this paper, both 'ECAPA' and 'Xvector' refer to `stage-1` attribute classifiers that use *ECAPA* and *Xvector embeddings*, respectively, as inputs.

Attributes Classification Technique (AC): In the second approach, we directly train `stage-1` attribute classifiers using Mel Frequency Cepstral Coefficients (MFCCs) as the input. The architecture details are in Section 6.4. This approach is designed for direct analysis of audio signals, allowing our classifiers to efficiently process the raw audio data.

These two approaches allow for flexibility within our classifiers, catering to different requirements of analysis —- whether it needs analysis using sophisticated speaker embeddings or direct classification using raw audio features.

6.2.2 Stage-2: Attribute-based SV

After obtaining `stage-1` attribute classifiers, we start the second stage of training SV using attributes, shown in Figure 6.1, that comprises the steps described below.

Creation and Labeling of Audio Pairs: We begin by generating a collection of audio pairs, categorizing them into positive pairs (from the same speaker) labeled as 1, and negative pairs (from different speakers) labeled as 0. This setup forms the groundwork for training our explainable models and is further discussed in Experiment Setup Section 6.4.

Attribute Classifier Processing: Each audio in the pairs is analyzed using the `stage-1` attribute classifiers to extract attribute labels and their corresponding probability vectors from the last layer of the classifier. A similarity vector for each pair is then constructed, with dimensions equal to the number of selected attributes, to quantify the attribute-wise similarity. To compute the similarity vector, we first need to determine the similarity of the corresponding attribute labels between the audio pairs. To address this, we propose two methods.

- **Hard Label Similarity:** In this approach, the similarity is considered binary: 0 for different labels, 1 for same labels.
- **Softmax Label Similarity:** Rather than a strict comparison with hard labels, this method compares probability vectors from the classifers' last layer. A probability vector reflects the likelihood of each potential class. For example, in nationality classification, a probability vector indicates the likelihood of the speaker belonging to each of several countries. We then calculate the cosine similarity between these vectors for each attribute.

Calculating Similarity and Constructing Vectors: We then compute the similarity using either hard labels or softmax labels for each attribute, constructing a similarity vector $[s_{gen}, s_{nat}, s_{age}, s_{pro}]$ for each audio pair. Here, s_{gen}, s_{nat}, s_{age}, and s_{pro} represent the similarity scores for our chosen attributes: gender, nationality, age, and profession, respectively, described in the following section.

Training with Machine Learning Models: Each of the aforementioned similarity vectors serves as a training point that is fed into a machine learning model for training. We explore several machine learning models, namely Linear Regression, Random Forest and the Logistic Regression.

In addition to these simple models, we also explore a simple neural network based model with two layers—a hidden layer and an output layer. The neural network takes the similarity vectors as input and processes them through the hidden layer, using a sigmoid activation function to capture complex relationships between attributes, finally outputting a similarity score for the audio pair.

6.3 Attributes and Datasets

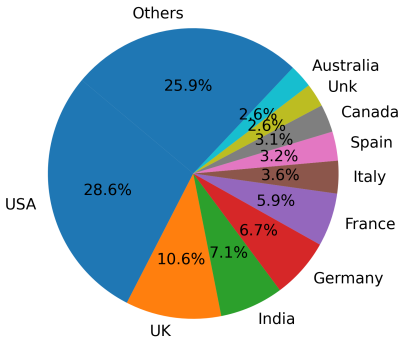


Figure 6.2: The nationality distribution of the 5994 speakers in the VoxCeleb 2 training set. Only the top 10 most frequent nationalities are shown individually in this figure, with the rest grouped as 'Others'.

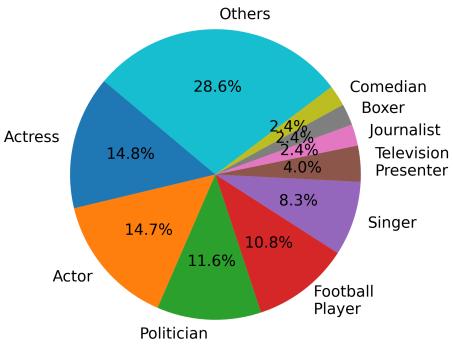


Figure 6.3: The profession distribution of the 5994 speakers in the VoxCeleb 2 training set. Only the top 10 most frequent professions are shown individually in this figure, with the rest grouped as 'Others'.

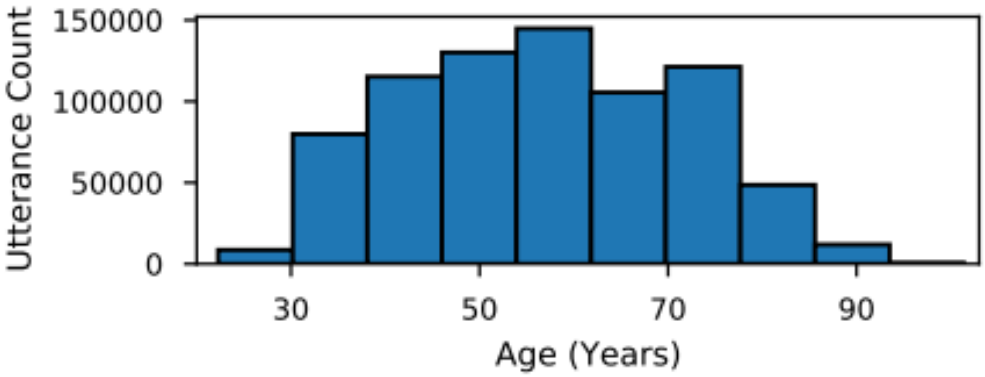


Figure 6.4: The age distribution of utterances in the SCOTUS corpus, split into 10 bins.

Figure 6.5: Attribute distributions across datasets used in this thesis.

Our choice of group-level attributes is driven by their availability in datasets and their role in speaker verification. However, group-level attributes, like nationality, age and gender, are rarely available together. For instance, widely used SV datasets like VoxCeleb provide only gender. Our key insight is that we can leverage these attributes in an ah-hoc manner, when attribute labels can be obtained for a particular dataset.

Initially, we selected gender, age, and nationality as group-level attributes for speaker verification, based on research showing their influence and biases in SV systems [142, 143, 144, 145]. These attributes are also widely used in forensic phonetics [136, 137] for identifying speakers in various contexts.

To further enrich the set of attributes, we introduced *profession*, a previously unexplored speaker labelling. What makes profession particularly valuable is that, like gender and nationality, it meets the criteria of being relevant to speaker characterization and readily available from external sources of information for celebrities in Voxceleb dataset. In this context, profession offers a dimension of speaker characterization that we expect to be independent of the other attributes listed above. Whilst we do not claim that a fine-grained information on a speaker's profession can be discerned from their voice in general contexts, such a labelling serves as a proxy for sociolect, influencing lexicon, syntax, and style—critical elements in forensic phonetics [136, 137] shaped by education and professional background. For example, broadcasters often develop a clear, articulate “broadcast voice" tailored for maximum intelligibility and audience engagement. Similarly, pop singers emphasize vocal control, rhythm, and pitch to convey emotion and style, all of which are heavily shaped by the demands of their professions.

Further, the Voxceleb dataset is particularly suited to studying the effects of profession on speech, as it includes recordings of individuals performing in professional contexts—politicians delivering speeches, singers performing, and comedians presenting routines. To confirm our hypothesis that profession is representative of speaker characteristics, we verify in Section 6.6 that profession is predictable from audio input.

Datasets: We use VoxCeleb, augmented with babble, music, background noise, and reverberation following Kaldi methods [38], for gender, nationality, and profession, and the SCOTUS corpus for age. Labels for gender, nationality, and age are sourced from [142], and profession labels are derived from Wikipedia, covering 49 distinct categories such as politicians, actors, and singers. Figures 6.2, 6.3 and 6.4 display the distribution of these attributes. VoxCeleb2 has 125 nationality labels and is predominantly composed of speakers from the USA (28.6%) and the UK (10.6%), with significant representation from India, Germany, and France. Professionally, the dataset includes many actors, politicians, and sports celebrities. The SCOTUS corpus, as shown in the age distribution from [142], is concentrated in the middle-aged demographic, particularly between 40 and 70 years.

6.4 Experimental Setup

Stage-1: To train `stage-1` attribute classifiers with Xvector and ECAPA embeddings, we use two hidden layers with Leaky ReLU activation leading to class projection. For attribute classifiers (AC) using MFCCs, we employ several TDNN layers, a pooling layer, two fully connected layers, and a softmax layer for classification. Training parameters: Each `stage-1` classifier is trained for 100,000 iterations with a batch size of 256. We use stochastic gradient descent with a learning rate of 0.2 and momentum of 0.5 for optimization. Details can be found in the source code.

Stage-2: We randomly select 160 speakers from VoxCeleb2 to create 160,000 trials (80,000 positive and 80,000 negative) for training `stage-2` models. Unlike traditional SV methods that do not use pair-based training, our approach calculates similarity based on differences in attribute labels between pairs of speakers, which requires training to be pair-based. While this approach is different from traditional methods, it is important to clarify that our evaluations use the same test set, ensuring comparability of results.

Test Set: `Stage-1` attribute classifers and `stage-2` machine learning models are evaluated on the VoxCeleb 1 test set, which consists of 40 speakers and 37,720 trials.

6.5 Metrics

We use two metrics: Accuracy and Equal Error Rate (EER). **Accuracy** reflects the precision of `stage-1` attribute classifiers in identifying attributes correctly–the higher, the better. **EER** is a performance measure used in SV to determine the threshold value where the false acceptance rate (FAR) equals the false rejection rate (FRR). Lower is better.

6.6 Results and Discussions

We present the following evaluations on the VoxCeleb1 test set: 1. A comparison of the `stage-1` attribute classifiers and their impact on the SV task; 2. The effect of using Hard Labels vs Softmax Labels in `stage-2`; 3. An analysis of the relative importance of the four attributes for SV within our framework.

	Random	Xvector	ECAPA	AC
Gender	0.50	0.99	0.99	0.99
Nationality	0.32	0.72	0.70	0.75
Profession	0.13	0.56	0.65	0.67
Age	0.17	0.66	0.73	0.78

Table 6.1: Accuracy of three sets of `stage-1` attribute classifiers—Xvector, ECAPA, and AC—across four attributes (gender, nationality, profession, and age).

		Softmax Labels			Hard Labels			
	Groundtruth	Xvector	ECAPA	AC	Xvector	ECAPA	AC	Random
Linear Regression	0.15	0.22	0.21	0.20	0.36	0.26	0.25	0.50
Random Forest	0.15	0.22	0.18	0.21	0.27	0.23	0.26	0.50
Logistic Regression	0.15	0.21	0.20	0.20	0.29	0.23	0.23	0.50
Neural Network	0.15	0.21	0.18	0.20	0.36	0.23	0.25	0.50
Xvector-org	0.035							
ECAPA-org	0.018							

Table 6.2: EER of 4 `stage-2` machine learning models(Linear regression, Random Forest, Logistic Regression, Neural Network) using softmax labels and hard labels from three sets of `stage-1` attribute classifiers(Xvector, ECAPA, and AC).

	Random	Xvector	ECAPA	AC	Groundtruth
Gender	0.50	0.40	0.34	0.34	0.36
profession	0.50	0.29	0.26	0.26	0.25
Nationality	0.50	0.36	0.34	0.35	0.27
Age	0.50	0.41	0.38	0.41	
All	0.50	0.20	0.18	0.21	0.15

Table 6.3: EER when using gender-only, profession-only, nationality-only, age-only and all softmax labels from three sets of `stage-1` attribute classifiers. When all softmax labels are utilized, Random Forest is employed as the `stage-2` model.

6.6.1 Comparison of `Stage-1` attribute classifiers:

Table 6.1 compares the accuracies of the `stage-1` attribute classifiers–vector, ECAPA, and AC–across gender, nationality, profession, and age. The `Random` column serves as a baseline, sampling attributes based on label distribution. All classifiers significantly outperform random guessing across all attributes. For example, for profession, random accuracy is 0.13, while Xvector, ECAPA, and AC achieve 0.56, 0.65, and 0.67, respectively, demonstrating that profession is indeed predictable from voice recordings.

Delving deeper into the comparison between Xvector, ECAPA, and AC, we notice that AC consistently outperforms the others across all attributes. This is because AC is trained directly on MFCCs, preserving more information than the pretrained embeddings used by Xvector and ECAPA. We also find that ECAPA generally surpasses Xvector, likely due to the use of more informative embeddings, highlighting its effectiveness in capturing relevant speaker attributes.

Next, we explore how these classifiers support `stage-2` machine learning models. Table 6.2 reports the EER of four `stage-2` models—Linear Regression, Random Forest, Logistic Regression, and Neural Network—using softmax or hard labels from the `stage-1` classifiers: Xvector, ECAPA, and AC. The `Groundtruth` column shows results using fully accurate labels, representing the best possible performance. The `Random` column, serving as a baseline, reflects outcomes from random guessing for comparison.

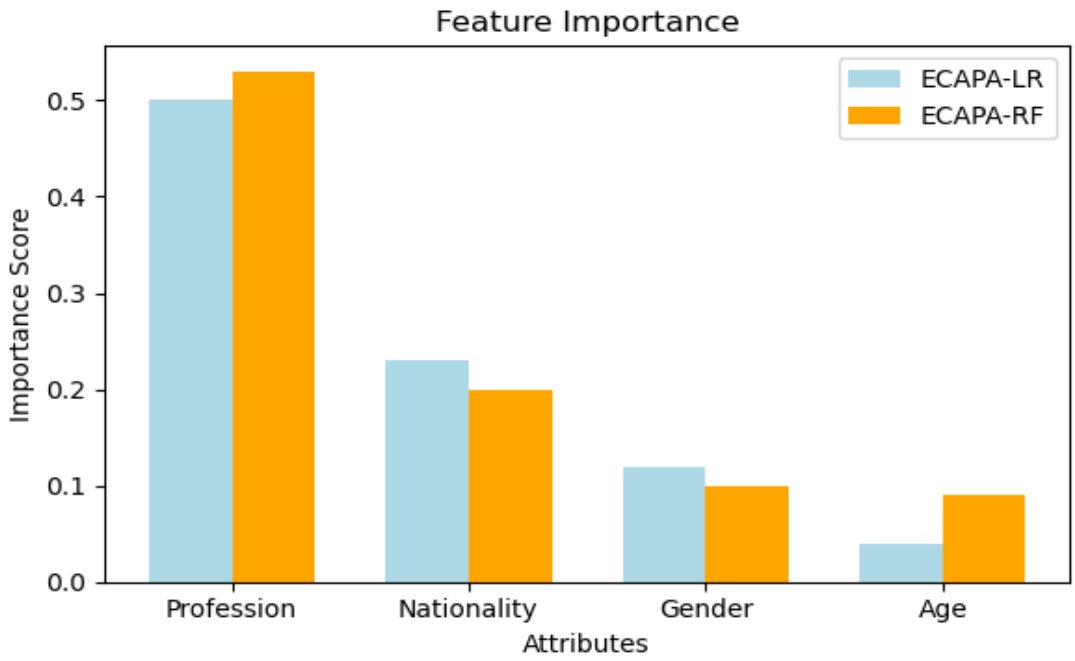


Figure 6.6: Feature Importance Scores from ECAPA-Linear Regression(ECAPA-LR) and ECAPA-Random Forest(ECAPA-RF), using softmax labels.

The red shaded area of Table 6.2 shows similarity vector computed using softmax labels and the blue shaded area is using hard labels. We find using the ECAPA classifier with softmax labels for similarity followed by Random Forest or Neural Network model achieves the best performance, with EER of 0.18. This outcome is slightly better than those obtained using Linear Regression and Logistic Regression, likely due to the ability of Random Forest and Neural Networks in capturing complex non-linear interactions among attributes. Also, it significantly surpasses the `Random` guess baseline of 0.50 and closely approaches the optimal `Groundtruth` result of 0.15, which shows the effectiveness of the ECAPA-Random Forest or ECAPA-Neural Network explainable attribute-based SV model. It is worth noting that performance of AC is comparable to ECAPA across all four `stage-2` machine learning models, owing to the similar accuracies in attribute prediction observed in `stage-1`, as seen earlier in Table 6.1. In contrast, Xvector classifiers slightly underperform ECAPA and AC across all `stage-2` machine learning models. This also aligns with our findings for `stage-1` attribute accuracy in Table 6.1 with Xvector having the least prediction accuracy.

It is worth noting that initially, we assume that relying solely on a limited set of group-level attributes would result in poor performance. However, our results show that the model performs reasonably well, achieving an EER of 0.18, which is close to the best achievable performance of 0.15 with the current set of attributes. Despite lower EERs observed with

SOTA models like ECAPA-org (0.035) and Xvector-org (0.018) that use complex audio embeddings, our findings indicate that simple group-level attributes contribute significantly to SV. Considering these attributes is valuable for designing a fully transparent and explainable model.

Looking ahead, expanding the attribute set to include more features will offer the potential to improve performance. This will enhance the explainability of our approach and offer deeper insights into the factors influencing SV outcomes.

In summary, regardless of which `stage-2` machine learning models is employed, `stage-1` classifiers AC and ECAPA demonstrate comparable performance. Notably, ECAPA, when followed by either a Random Forest or a Neural Network, exhibits a slight advantage. Based on these results, we recommend ECAPA as the preferred `stage-1` attribute classifier.

6.6.2 Softmax versus Hard Label Similarity in Stage-2:

Across all `stage-1` attribute classifiers and `stage-2` machine learning models, we find using softmax labels produces lower EER than hard labels. We find the nuanced category probabilities provided by softmax labels help `stage-2` models compute more accurate similarity scores, as reflected in the lower EER in Table 6.2.

6.6.3 Importance of Attributes

We evaluate the EER when using a single attribute—gender-only, profession-only, nationality-only, and age-only softmax labels from the three `stage-1` classifiers—comparing them against random guessing based on label distribution and ground truth performance. Results are shown in Table 6.3. For comparison, we also present the EERs using `All` attributes with the best-performing `stage-2` Random Forest model.

Examining the ground truth EERs in Table 6.3, we find profession (0.25) and nationality (0.27) outperform gender (0.36), suggesting that, with accurate predictions, profession and nationality are great indicators of speaker identity.

With softmax labels from `stage-1` attribute classifiers, profession consistently outperforms other attributes across all `stage-1` classifiers. This shows profession's value in the SV task. Afer profession, nationality and gender have comparable importance based on their EERs. We believe the higher EER of nationality compared to profession is due to the sparsity of nationalities in the Voxceleb1 test dataset. Gender, with just two labels, offers less discriminative power but remains relevant to SV. Conversely, age is found to be less impactful (Xvector: 0.41, ECAPA: 0.38, AC: 0.41). `Stage-2` machine learning models, such as Linear Regression and Random Forest, provide importance scores for the four attributes. For Linear Regression, these are based on the absolute values of the model's weights. Figure 6.6 shows the feature importance scores for the four attributes based on ECAPA-Linear Regression and

ECAPA-Random Forest using softmax labels. both models rank *Profession* highest, followed by *Nationality*, *Gender*, and *Age*. **This trend aligns with Table 6.3 and highlights the interpretability of our approach, as we can clearly see which and how attributes drive predictions.**

6.7 Limitation

A limitation of this study is the use of profession as one of the concept attributes. Since the VoxCeleb dataset is primarily composed of celebrity interviews, speech style may be influenced by the speaker's profession, such as actors or politicians, where performance or rhetorical style plays a role. This raises concerns about whether the attribute generalises to non-celebrity voices, where profession may not strongly shape speech characteristics.

Another limitation arises from the fact that the attributes modelled in the bottleneck layer, such as gender, age, and profession, are often considered sensitive or protected characteristics in the context of fairness and responsible AI. While exposing these attributes enables greater transparency in understanding model decisions, it also introduces risks of misuse, for example by reinforcing social biases or enabling unintended profiling. This tension highlights an inherent challenge in intrinsic explainability: the very attributes that make models more interpretable can also raise fairness concerns. Addressing this trade-off will require careful consideration of ethical guidelines and the design of safeguards in future work.

6.8 Conclusion

Our research takes a significant step towards understanding the influence of various attributes–age, gender,nationality and profession–on speaker verification performance. We develop a two-stage explainable attribute-based framework, starting with training stage-1 attribute classifier followed by using stage-2 machine learning models to verify speakers using these attributes. Evaluated on the Voxceleb1 test dataset, we find profession and nationality have a considerable influence on SV performance, followed by gender and age. In the future, we aim to expand these attributes seeking to improve SV performance while providing transparency and explainability, promising a new direction for SV systems.

Chapter 7

Conclusion

7.1 Overview

This thesis has investigated two challenges of robustness and explainability in speech AI, focusing on Automatic Speech Recognition (ASR) and Speaker Verification (SV). While deep learning has delivered substantial performance improvements in these tasks, it has also introduced vulnerabilities to adversarial manipulation and reduced transparency of decision-making. Addressing these issues is increasingly important for both academic research and real-world deployment, as speech technologies are widely integrated into safety-critical applications.

Guided by three overarching research questions, this thesis explored how ASR systems respond to adversarial perturbations (RQ1), how post-hoc explainability methods can be applied and evaluated in ASR (RQ2), and whether intrinsic explainability can be achieved in SV through the use of human-understandable attributes (RQ3). The following sections summarise the answers to these research questions, highlight the key limitations of the work, and outline directions for future research.

7.2 Addressing the Research Questions

RQ1: How do ASR systems behave under adversarial perturbations?

Chapter 3 introduced SPAT, a psychoacoustically motivated black-box adversarial attack framework. SPAT demonstrated that ASR systems, including both open-source and commercial models, can be severely degraded by perturbations that are imperceptible to human listeners. Experiments across multiple attack generation methods and frame selection strategies showed that high word error rates (WER) can be achieved while maintaining perceptual similarity to the original signal.

These results demonstrate that current ASR systems remain fragile to adversarial manipulation. For developers, this underscores the need to evaluate robustness not only on clean benchmarks but also against adversarially generated audio, and to incorporate perceptual and semantic plausibility into robustness assessment.

RQ2: How effective are post-hoc explainability methods when applied to ASR, and how can their reliability be systematically evaluated?

Chapters 4 and 5 examined perturbation-based post-hoc explainability in ASR. In Chapter 4, the `X-ASR` framework was introduced to apply LIME, SFL, and Causal to ASR outputs. Experiments showed that SFL and causa produced more compact and consistent explanations than LIME, though all methods exhibited sensitivity to perturbation granularity and similarity metrics. These results revealed both the potential and the fragility of applying post-hoc explainability methods to speech tasks.

Chapter 5 addressed the lack of systematic evaluation by introducing a benchmarking framework based on phoneme recognition (PR). By exploiting the availability of ground-truth phoneme alignments in TIMIT, explanations were quantitatively assessed. Results showed that the proposed LIME-TS variant achieved high reliability, containing ground-truth phoneme segments in 96% of cases, while SFL and causal attribution maintained superior compactness and stability. This established one of the first structured methods for quantitatively comparing explanation techniques in ASR.

Overall, the findings provide a comprehensive answer to RQ2: post-hoc explainability can be successfully adapted to ASR, but its reliability depends on careful design choices, and quantitative benchmarking is essential to move beyond qualitative inspection.

RQ3: Can intrinsic explainability be achieved in speaker verification?

Chapter 6 explored intrinsic explainability using Concept Bottleneck Models (CBMs) for speaker verification. Unlike post-hoc methods, CBMs enforce an intermediate layer where representations are explicitly tied to interpretable attributes. This allowed us to examine how factors such as accent, gender, and profession contribute to verification outcomes. Experiments demonstrated that SV predictions can indeed be mediated through such attributes, providing transparent explanations of model behaviour.

However, limitations were also identified. The use of profession as an attribute is tied to the VoxCeleb dataset, where celebrity status and public speaking style may confound the relationship between profession and vocal characteristics, limiting generalisability to everyday speech. Furthermore, attributes such as gender, age, and profession are often considered sensitive, raising fairness concerns when exposed in model decisions. These findings illustrate both the promise and the ethical challenges of intrinsic explainability in SV.

In sum, the results of Chapter 6 show that intrinsic explainability is feasible in SV and complements the post-hoc analyses of ASR, but they also highlight the need to carefully balance interpretability with fairness considerations.

7.3 Limitations

Despite the contributions presented, this thesis has several limitations. The experimental work was restricted to datasets such as TIMIT and VoxCeleb, chosen for their controlled conditions and availability of ground-truth alignments. While these choices enabled precise analysis, they limit the immediate generalisability of the findings to modern large-scale or end-to-end architectures. Methodologically, the focus on perturbation-based explainability provided a coherent foundation but left other families of approaches unexplored. Finally, the attribute-based modelling in speaker verification drew on dataset-specific categories such as profession, which may not transfer to everyday speech and intersects with sensitive characteristics, raising questions of fairness. These limitations do not diminish the contributions but set boundaries on their applicability, and they point naturally towards future directions.

7.4 Future Work

There are several promising avenues for extending this work. One direction concerns the adaptation of robustness and explainability techniques to state-of-the-art speech architectures. As transformer-based and foundation models become dominant in ASR, it will be essential to revisit both adversarial vulnerability and the applicability of post-hoc explanation methods in these settings. Such studies will help determine whether the patterns observed in traditional pipelines persist or evolve with scale and model complexity.

A second direction involves broadening the methodological scope. Beyond perturbation-based attribution, gradient-based saliency methods, prototype-driven reasoning, and disentangled representation learning may offer complementary insights. Systematic comparisons across these families could provide a more complete understanding of what makes explanations reliable and meaningful in speech AI. Equally important is the semantic evaluation of adversarial outputs: moving beyond word error rate towards user studies and task-driven assessments that capture how humans perceive and react to manipulated speech.

Intrinsic explainability also invites further development. The concept bottleneck approach demonstrated that speaker attributes can serve as interpretable mediators of verification decisions, but future work should explore attributes that generalise more broadly and are less entangled with sensitive characteristics. Fairness-aware or privacy-preserving adaptations of CBMs may help reconcile the tension between transparency and ethical responsibility.

Finally, extending explainability beyond general-purpose speech tasks into high-stakes domains remains a critical challenge. In healthcare, for example, explainability is indispensable for building trust with clinicians and patients, as speech is increasingly explored as a biomarker for neurological and mental health conditions. Embedding explainability frameworks into such applications will require not only technical adaptation but also interdisciplinary collaboration with clinicians, ethicists, and regulators to ensure safety, reliability, and societal acceptance.

7.5 Closing Remarks

This thesis has presented a comprehensive investigation of robustness and explainability in speech AI. Through adversarial attack studies, post-hoc explainability frameworks, and intrinsic concept-based modelling, it has shown both the vulnerabilities and the interpretability opportunities of ASR and SV. The work highlights that achieving robust and explainable speech AI requires not only technical innovation but also attention to usability, fairness, and deployment contexts. As speech technologies continue to advance and integrate into everyday life, the insights from this thesis provide a foundation for building systems that are both effective and trustworthy.

Appendix A

Chapter 3: Extension Results

A.1 RQ1: Comparison of Frame Selection Techniques

We present one-way Anova and Tukey's Honest Significant Difference (HSD) test(at 5% significance level) on `WER` and `Similarity` to compare our frame selection techniques.

A.1.1 `WER`: P-values for pairwise comparisons of `WERs` between frame selection techniques.

	Librispeech			Commonvoice		
	Deepspeech	Sphinx	Google	Deepspeech	Sphinx	Google
All vs Random	**0.001**	**0.001**	**0.001**	**0.011**	0.07	**0.31**
All vs Important	**0.043**	**0.001**	0.06	0.40	0.9	0.43
Important vs Random	**0.001**	**0.001**	**0.006**	0.35	0.23	0.9

Table A.1: P-values for pairwise comparison of `WER` achieved by frame selection methods (using `GL` attack generation).

	Librispeech			Commonvoice		
	Deepspeech	Sphinx	Google	Deepspeech	Sphinx	Google
All vs Random	**0.001**	**0.001**	0.23	0.58	**0.001**	0.9
All vs Important	**0.036**	**0.001**	0.28	0.51	**0.001**	0.9
Important vs Random	**0.03**	**0.032**	0.9	0.07	0.9	0.9

Table A.2: P-values for pairwise comparison of `WER` achieved by frame selection methods (using `DE` attack generation).

	Librispeech			Commonvoice		
	Deepspeech	Sphinx	Google	Deepspeech	Sphinx	Google
All vs Random	**0.001**	**0.001**	0.59	0.58	**0.03**	0.9
All vs Important	**0.001**	**0.001**	0.80	0.9	**0.04**	0.87
Important vs Random	0.228	**0.01**	0.85	0.76	0.8	0.9

Table A.3: P-values for pairwise comparison of `WER` achieved by frame selection methods (using `OP` attack generation).

A.1.2 `Similarity`: P-values for pairwise comparisons of `Similarity` between frame selection techniques.

	Librispeech	Commonvoice
Random VS All	**0.01**	**0.014**
Important VS All	0.57	0.9
Random VS Important	0.09	0.06

Table A.4: P-values for pairwise comparison of `Similarity` achieved by frame selection methods (using `GL` attack generation).

	Librispeech	Commonvoice
Random VS All	**0.001**	**0.001**
Important VS All	**0.001**	**0.001**
Random VS Important	0.34	0.9

Table A.5: P-values for pairwise comparison of `Similarity` achieved by frame selection methods (using `DE` attack generation).

	Librispeech	Commonvoice
Random VS All	**0.001**	**0.001**
Important VS All	**0.001**	**0.001**
Random VS Important	0.09	0.11

Table A.6: P-values for pairwise comparison of `Similarity` achieved by frame selection methods (using `OP` attack generation).

Tables A.6, A.5 and A.4 in Section A.1.2 do not show different ASRs as the adversarial attacks are agnostic to the ASR used.

A.1.3 Pareto Front: Number of non-dominated samples for three frame selection techniques

Table A.7 and A.8 compares frame selection configurations for a fixed attack generation in terms of number of non-dominated samples on two datasets. Column heading in the table shows the fixed parameter; we fix one attack generation at a time and compare frame selection configurations.

	Deepspeech			Sphinx			Google		
	GL	OP	DE	GL	OP	DE	GL	OP	DE
All	4	3	1	7	5	3	5	2	5
Random	3	7	12	5	7	9	6	5	8
Important	4	9	17	7	9	13	6	9	9

Table A.7: Number of non-dominated samples for frame selection techniques using different attack and ASRs on **Commonvoice**

	Deepspeech			Sphinx			Google		
	GL	OP	DE	GL	OP	DE	GL	OP	DE
All	3	3	3	7	6	4	1	2	2
Random	8	4	5	6	3	6	7	6	8
Important	9	5	7	7	6	6	8	7	13

Table A.8: Number of non-dominated samples for frame selection techniques using different attack and on ASRs on **librispeech**

A.2 RQ2: Comparison of Attack Generation Techniques

We present one-way Anova and Tukey's Honest Significant Difference (HSD) test(at 5% significance level) on `WER` and `Similarity` to compare our attack generation techniques.

A.2.1 `WER`: P-values for pairwise comparisons of `WER`s between frame selection techniques.

	Librispeech			Commonvoice		
	Deepspeech	Sphinx	Google	Deepspeech	Sphinx	Google
GL vs OP	**0.001**	**0.001**	**0.001**	**0.009**	**0.001**	0.81
GL vs DE	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	0.79
OP vs DE	0.55	0.66	0.9	0.75	0.63	0.81

Table A.9: P-values for pairwise comparison of `WER` achieved by attack generation methods (using `Important` frames).

	Librispeech			Commonvoice		
	Deepspeech	Sphinx	Google	Deepspeech	Sphinx	Google
GL vs OP	**0.001**	**0.001**	**0.007**	**0.009**	**0.001**	0.9
GL vs DE	**0.001**	**0.001**	**0.001**	**0.006**	**0.001**	0.9
OP vs DE	0.9	0.66	0.9	0.9	0.21	0.9

Table A.10: P-values for pairwise comparison of `WER` achieved by attack generation methods (using `Random` frames).

	Librispeech			Commonvoice		
	Deepspeech	Sphinx	Google	Deepspeech	Sphinx	Google
GL vs OP	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	0.189
GL vs DE	**0.001**	**0.001**	**0.001**	**0.002**	**0.001**	0.05
OP vs DE	**0.04**	**0.03**	0.60	0.9	0.58	0.818

Table A.11: P-values for pairwise comparison of `WER` achieved by attack generation methods (using `All` frames).

A.2.2 `Similarity`: P-values for pairwise comparisons of `Similarity` between attack generation techniques.

	Librispeech	Commonvoice
OP VS GL	**0.001**	**0.001**
DE VS GL	**0.001**	**0.001**
DE VS OP	0.1	**0.001**

Table A.12: P-values for pairwise comparison of `Similarity` achieved by attack generation methods (using `Important` frames).

	Librispeech	Commonvoice
OP VS GL	**0.001**	**0.001**
DE VS GL	**0.001**	**0.001**
DE VS OP	0.38	**0.001**

Table A.13: P-values for pairwise comparison of `Similarity` achieved by attack generation methods (using `Random` frames).

	Librispeech	Commonvoice
OP VS GL	**0.001**	**0.001**
DE VS GL	**0.001**	**0.001**
OP VS DE	0.06	0.56

Table A.14: P-values for pairwise comparison of `Similarity` achieved by attack generation methods (using `All` frames).

A.2.3 Pareto Front: Number of non-dominated samples for three attack generation techniques

Table A.15 and A.16 compares attack generation configurations for a fixed frame selection in terms of number of non-dominated samples on two datasets.

	Deepspeech			Sphinx			Google		
	All	Random	Important	All	Random	Important	All	Random	Important
GL	2	1	2	7	7	10	1	1	5
OP	10	0	5	8	1	2	11	3	5
DE	6	17	16	9	23	25	8	20	12

Table A.15: Number of non-dominated samples for attack generation techniques using different frame selection techniques and ASRs on **Commonvoice**

	Deepspeech			Sphinx			Google		
	All	Random	Important	All	Random	Important	All	Random	Important
GL	4	2	4	7	7	6	3	3	2
OP	8	3	7	8	1	6	8	1	5
DE	1	7	9	4	7	8	2	4	8

Table A.16: Number of non-dominated samples for attack generation techniques using different frame selection techniques and ASRs on **Librispeech**

A.3 RQ4: Comparison with Abdullah et al.

Results comparing our attack with Abdullah et al. on Librispeech dataset is shown in Table A.18. We present one-way Anova and Tukey's Honest Significant Difference (HSD) test (at 5% significance level) on `WER` and `Similarity` in Tables A.17 and A.19 for Commonvoice and Librispeech datasets, respectively.

A.3.1 P-values for the comparison of `WER` and `Similarity` between our approach and Abdullah et al. on Commonvoice dataset.

	Similarity	WER on Deepspeech	WER on Sphinx	WER on Google
`OP+All` vs Abdullah's work	**0.041**	**0.026**	**0.037**	**0.001**
`OP+Important` vs Abdullah's work	**0.001**	0.66	0.08	**0.003**
`DE+All` vs Abdullah's work	0.79	**0.027**	**0.013**	**0.001**
`DE+Important` vs Abdullah's work	**0.001**	0.88	0.06	**0.001**

Table A.17: P-values for pairwise comparison of `Similarity` and `WER` achieved by Abdullah et al, against `OP+All`, `OP+Important`, `DE+All`, `DE+Important` on Commonvoice dataset.

A.3.2 Comparison with Abdullah et al. on Librispeech Dataset

Technique	Time	Similarity	Success rate			WER			Detection score
			Deepspeech	Sphinx	Google	Deepspeech	Sphinx	Google	
Abdullah	22 seconds	2.6	76%	86%	50%	0.10	0.18	0.06	0.40
OP	3.5 seconds	3.65	95%	96.5%	97.5%	0.17	0.28	0.20	0.14
DE	2.5 seconds	3.72	91%	94%	95.5%	0.11	0.19	0.20	0.11

Table A.18: Comparison of `OP`, `DE` with Abdullah et al. with respect to generation time for per adversarial audio sample, `Similarity` to original audio samples,`WER`, `Success Rate` and `Detection score` against defense system in attacking all three ASRs on Librispeech dataset

A.3.3 P-values for the comparison of WER and Similarity between our approach and Abdullah et al. on Librispeech dataset.

	Similarity	WER on Deepspeech	WER on Sphinx	WER on Google
OP+All vs Abdullah's work	**0.001**	**0.001**	**0.012**	**0.001**
OP+Important vs Abdullah's work	**0.001**	0.38	0.56	**0.001**
DE+All vs Abdullah's work	**0.009**	0.077	0.072	**0.001**
DE+Important vs Abdullah's work	**0.009**	0.13	0.9	**0.001**

Table A.19: P-values for comparison of Similarity and WER achieved by Abdullah et al. against OP+All, OP+Important, DE+All, DE+Important on Librispeech dataset.

A.4 Listening Test

To complement the quantitative evaluation, we conducted a small-scale listening test to assess the perceptual realism of the adversarial examples. A total of 20 participants were presented with 100 pairs of original and adversarial audio samples in randomised order and asked to identify which one had been modified.

The results indicated that participants were only slightly better than chance (average accuracy below 55%), suggesting that the perturbations were largely imperceptible to human listeners. This finding reinforces the psychoacoustic motivation of SPAT: attacks can remain undetectable to human ears while still significantly disrupting ASR performance.

While this study was limited in scale and conducted under controlled conditions, it provides supporting evidence that the adversarial examples generated are realistic and perceptually natural. A more comprehensive user study remains an avenue for future work.

Bibliography

[1] Qing Wang, Pengcheng Guo, and Lei Xie. Inaudible adversarial perturbations for targeted attack in speaker recognition. *CoRR*, abs/2005.10637, 2020. URL https://arxiv.org/abs/2005.10637.

[2] Hadi Abdullah, Muhammad Sajidur Rahman, Washington Garcia, Logan Blue, Kevin Warren, Anurag Swarnim Yadav, Tom Shrimpton, and Patrick Traynor. Hear "no evil", see "kenansville": Efficient and transferable black-box attacks on speech recognition and voice identification systems, 2019.

[3] Nicholas Carlini and David A. Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. *CoRR*, abs/1801.01944, 2018. URL http://arxiv.org/abs/1801.01944.

[4] Shehzeen Hussain, Paarth Neekhara, Shlomo Dubnov, Julian J. McAuley, and Farinaz Koushanfar. Waveguard: Understanding and mitigating audio adversarial examples. *CoRR*, abs/2103.03344, 2021. URL https://arxiv.org/abs/2103.03344.

[5] Amazon. Alexa unveils new speech recognition, text-to-speech technologies, 2023. URL https://www.amazon.science/blog/alexa-unveils-new-speech-recognition-text-to-speech-technologies. Accessed: 2025-05-01.

[6] Apple. Voice trigger system for siri, 2023. URL https://machinelearning.apple.com/research/voice-trigger. Accessed: 2025-05-01.

[7] Zoom. Release notes for august 7, 2022, 2022. URL https://support.zoom.com/hc/en/article?id=zm_kb&sysparm_article=KB0071060. Accessed: 2025-05-01.

[8] Google. Google meet, 2023. URL https://meet.google.com/. Accessed: 2025-05-01.

[9] Athreon. Mobile dictation apps for ios and android, 2023. URL https://www.athreon.com/speech-to-text-mobile-apps/. Accessed: 2025-05-01.

[10] The Guardian. Hsbc rolls out voice and touch id security for bank customers, 2016. URL https://www.theguardian.com/business/2016/feb/19/hsbc-rolls-out-voice-touch-id-security-bank-customers. Accessed: 2025-05-01.

[11] Microsoft. What's new in azure ai speech?, 2023. URL https://learn.microsoft.com/en-us/azure/ai-services/speech-service/releasenotes. Accessed: 2025-05-01.

[12] Google. Turn on voice recognition with voice match - android - google help, 2023. URL https://support.google.com/assistant/answer/9071681?co=GENIE.Platform%3DAndroid&hl=en. Accessed: 2025-05-01.

[13] R. Botros et al. Automated handling of emergency calls. *Journal of Emergency Services*, 10(2):123–130, 2022.

[14] M. Jessen. Speaker profiling and forensic voice comparison. In M. Coulthard and A. Johnson, editors, *The Routledge Handbook of Forensic Linguistics*, pages 382–399. Routledge, 2021.

[15] G. Deshpande et al. Vocal biomarker predicts fatigue in people with covid-19. *BMJ Open*, 12(11):e062463, 2023.

[16] T. Koizumi et al. Vocal acoustic features as potential biomarkers for identifying depression. *Frontiers in Digital Health*, 2:45–50, 2023.

[17] Regulation (eu) 2024/1689 of the european parliament and of the council of 13 march 2024 laying down harmonised rules on artificial intelligence (artificial intelligence act), 2024. URL https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32024R1689. Accessed: 2025-05-01.

[18] Organisation for Economic Co-operation and Development. Recommendation of the council on artificial intelligence, 2022. URL https://legalinstruments.oecd.org/en/instruments/OECD-LEGAL-0449. Accessed: 2025-05-01.

[19] National Institute of Standards and Technology. Artificial intelligence risk management framework (ai rmf 1.0), 2023. URL https://www.nist.gov/itl/ai-risk-management-framework. Accessed: 2025-05-01.

[20] Muhammad A. Shah, David Solans Noguero, Mikko A. Heikkilä, and Nicolas Kourtellis. Speech robust bench: A robustness benchmark for speech recognition. *arXiv preprint arXiv:2403.07937*, 2024. URL https://arxiv.org/abs/2403.07937.

[21] Moustapha Cisse, Yossi Adi, Natalia Neverova, and Joseph Keshet. Houdini: Fooling deep structured prediction models, 2017.

[22] Xuejing Yuan, Yuxuan Chen, Yue Zhao, Yunhui Long, Xiaokang Liu, Kai Chen, Shengzhi Zhang, Heqing Huang, Xiaofeng Wang, and Carl A. Gunter. Commandersong: A systematic approach for practical adversarial voice recognition. *CoRR*, abs/1801.08535, 2018. URL http://arxiv.org/abs/1801.08535.

[23] F. R. Goss, L. Zhou, and S. G. Weiner. Incidence of speech recognition errors in the emergency department. *International Journal of Medical Informatics*, 93:70–73, 2016. doi: 10.1016/j.ijmedinf.2016.05.005. URL https://www.sciencedirect.com/science/article/pii/S1386505616300909.

[24] Mozilla Foundation. You don't sound like what we're looking for: A fairness audit of automatic speech recognition systems, 2022. URL https://foundation.mozilla.org/en/blog/speech-recognition-fairness-audit/. Accessed: 2025-05-01.

[25] Robert Booth. Blind people excluded from benefits of ai, says charity, 2023. URL https://www.theguardian.com/society/2023/dec/25/blind-people-excluded-from-benefits-of-ai-says-charity. Accessed: 2025-05-01.

[26] Yao Qin, Nicholas Carlini, Ian Goodfellow, Garrison Cottrell, and Colin Raffel. Imperceptible, robust, and targeted adversarial examples for automatic speech recognition, 2019.

[27] Rohan Taori, Amog Kamsetty, Brenton Chu, and Nikita Vemuri. Targeted adversarial examples for black box audio systems, 2019.

[28] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y. Ng. Deep speech: Scaling up end-to-end speech recognition, 2014.

[29] Paul Lamere, Philip Kwok, William Walker, Evandro B Gouvêa, Rita Singh, Bhiksha Raj, and Peter Wolf. Design of the cmu sphinx-4 decoder. In *Interspeech*. Citeseer, 2003.

[30] Google. "google speech to text api", 2017. URL https://cloud.google.com/speech-to-text/.

[31] X. Wu, P. Bell, and A. Rajan. Explanations for automatic speech recognition. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2023. doi: 10.1109/ICASSP49357.2023.10094635.

[32] S. Becker, J. Vielhaben, M. Ackermann, K.-R. Müller, S. Lapuschkin, and W. Samek. Audiomnist: Exploring explainable artificial intelligence for audio analysis on a simple benchmark. volume 361, pages 418–428, 2024. doi: 10.1016/j.jfranklin.2023.11.038.

[33] P. Kumar, V. Kaushik, and B. Raman. Towards the explainability of multimodal speech emotion recognition. In *Interspeech 2021*, pages 1748–1752, 2021. doi: 10.21437/Interspeech.2021-1718.

[34] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. *CoRR*, abs/1602.04938, 2016. URL http://arxiv.org/abs/1602.04938.

[35] Youcheng Sun, Hana Chockler, Xiaowei Huang, and Daniel Kroening. Explaining deep neural networks using spectrum-based fault localization. *CoRR*, abs/1908.02374, 2019. URL http://arxiv.org/abs/1908.02374.

[36] Hana Chockler, Daniel Kroening, and Youcheng Sun. Compositional explanations for image classifiers. *CoRR*, abs/2103.03622, 2021. URL `https://arxiv.org/abs/2103.03622`.

[37] Andreas Madsen, Siva Reddy, and Sarath Chandar. Post-hoc interpretability for neural nlp: A survey. *arXiv preprint arXiv:2108.04840*, 2021.

[38] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, number CONF. IEEE Signal Processing Society, 2011.

[39] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett. DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1. NASA STI/Recon Technical Report N, February 1993.

[40] Li Deng and Douglas O'Shaughnessy. *Speech Processing: A Dynamic and Optimization-Oriented Approach*. Marcel Dekker, 2003.

[41] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 2016. URL `https://proceedings.mlr.press/v48/amodei16.html`.

[42] Lawrence R. Rabiner and Biing-Hwang Juang. *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[43] Steven Davis and Philip Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366, 1980. doi: 10.1109/TASSP.1980.1163420.

[44] Jerry D. Markel and Alan H. Gray. *Linear Prediction of Speech*. Springer, 1976.

[45] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460, 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/92d1e1eb1cd6f9fba3227870bb6d7f07-Abstract.html`.

[46] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[47] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Brian Kingsbury, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[48] Peter Ladefoged and Keith Johnson. *A Course in Phonetics*. Wadsworth, Cengage Learning, Boston, MA, 6th edition, 2012. ISBN 978-1-111-35015-9.

[49] Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn. Speaker verification using adapted gaussian mixture models. *Digital signal processing*, 10(1-3):19–41, 2000.

[50] John HL Hansen and Taufiq Hasan. Speaker recognition: Challenges, opportunities, and trends. *IEEE Signal Processing Magazine*, 32(6):106–125, 2015.

[51] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798, 2011.

[52] Tomi Kinnunen and Haizhou Li. An overview of text-independent speaker recognition: From features to supervectors. *Speech Communication*, 52(1):12–40, 2010.

[53] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust dnn embeddings for speaker recognition. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5329–5333. IEEE, 2018.

[54] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. *arXiv preprint arXiv:2005.07143*, 2020.

[55] Shang-Hua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip Torr. Res2net: A new multi-scale backbone architecture. *IEEE transactions on pattern analysis and machine intelligence*, 43(2):652–662, 2021.

[56] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR 2018*, pages 7132–7141, 2018.

[57] Feng Wang, Weiyang Cheng, Weiyang Liu, and Haijun Liu. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930, 2018.

[58] Zachary C. Lipton. The mythos of model interpretability. In *ICML Workshop on Human Interpretability in Machine Learning (WHI)*, 2016.

[59] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[60] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. *CoRR*, abs/1703.01365, 2017. URL `http://arxiv.org/abs/1703.01365`.

[61] Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? *arXiv preprint arXiv:1611.07450*, 2016.

[62] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. *CoRR*, abs/1704.02685, 2017. URL `http://arxiv.org/abs/1704.02685`.

[63] Lee Naish, Hua Jie Lee, and Kotagiri Ramamohanarao. A model for spectra-based software diagnosis. *ACM Transactions on software engineering and methodology (TOSEM)*, 20(3):1–32, 2011.

[64] Alexis Ross, Ana Marasovic, and Matthew E. Peters. Explaining NLP models via minimal contrastive editing (mice). *CoRR*, abs/2012.13985, 2020. URL `https://arxiv.org/abs/2012.13985`.

[65] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *arXiv preprint arXiv:1312.6034*, 2014.

[66] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[67] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[68] James A Jones, Mary Jean Harrold, and John Stasko. Visualization of test information to assist fault localization. In *Proceedings of the 24th International Conference on Software Engineering. ICSE 2002*, pages 467–477. IEEE, 2002.

[69] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harvard Journal of Law & Technology*, 31:841–887, 2017.

[70] Ramaravind Kommiya Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency (FAT* '20)*, pages 607–617. ACM, 2020.

[71] Joseph Y. Halpern and Judea Pearl. Causes and explanations: A structural-model approach — part 1: Causes. *CoRR*, abs/1301.2275, 2013. URL `http://arxiv.org/abs/1301.2275`.

[72] Alexandra Chouldechova and Aaron Roth. A snapshot of the frontiers of fairness in machine learning. *Communications of the ACM*, 63(5):82–89, 2020.

[73] Chaofan Chen, Oscar Li, Chaofan Tao, Alina Jade Barnett, Jonathan Su, and Cynthia Rudin. This looks like that: Deep learning for interpretable image recognition. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[74] Zichao Li, Yi Ding, Hema Raghavan, and William Yang Wang. Interpretable prototype selection for text classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1234–1244, 2021.

[75] Yao Ming, Huamin Qu, and Han-Wei Shen. Interpretable and steerable sequence learning via prototypes. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 903–913, 2019.

[76] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5338–5348. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/koh20a.html`.

[77] Anita Mahinpei, Justin Clark, Isaac Lage, Finale Doshi-Velez, and Weiwei Pan. Promises and pitfalls of black-box concept learning models. In *Proceedings of the ICML Workshop on Theoretic Foundation, Criticism, and Application Trend of Explainable AI*, 2021.

[78] An Yan, Yu Wang, Yiwu Zhong, Zexue He, Petros Karypis, Zihan Wang, Chengyu Dong, Amilcare Gentili, Chun-Nan Hsu, Jingbo Shang, and Julian McAuley. Robust and interpretable medical image classifiers via concept bottleneck models. *arXiv preprint arXiv:2310.03182*, 2023.

[79] Sapna Maheshwari. Burger king 'ok google'ad doesn't seem ok with google. *The New York Times. https://www. nytimes. com/2017/04/12/business/burger-king-tv-ad-google-home. html*, 2017.

[80] Shaun Nichols. Tv anchor says live on-air 'alexa, order me a dollhouse'-guess what happens next. *The Register*, 7, 2017.

[81] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[82] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[83] Alexey Kurakin, Ian Goodfellow, Samy Bengio, et al. Adversarial examples in the physical world, 2016.

[84] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017.

[85] Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David Wagner, and Wenchao Zhou. Hidden voice commands. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 513–530, Austin, TX, August 2016. USENIX Association. ISBN 978-1-931971-32-4. URL https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/carlini.

[86] Hiromu Yakura and Jun Sakuma. Robust audio adversarial example for a physical attack. *CoRR*, abs/1810.11793, 2018. URL http://arxiv.org/abs/1810.11793.

[87] Paarth Neekhara, Shehzeen Hussain, Prakhar Pandey, Shlomo Dubnov, Julian J. McAuley, and Farinaz Koushanfar. Universal adversarial perturbations for speech recognition systems. *CoRR*, abs/1905.03828, 2019. URL http://arxiv.org/abs/1905.03828.

[88] Moustafa Alzantot, Bharathan Balaji, and Mani B. Srivastava. Did you hear that? adversarial examples against automatic speech recognition. *CoRR*, abs/1801.00554, 2018. URL http://arxiv.org/abs/1801.00554.

[89] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210, 2015. doi: 10.1109/ICASSP.2015.7178964.

[90] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber. Common voice: A massively-multilingual speech corpus. In *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, pages 4211–4215, 2020.

[91] Yiqing Lin and Waleed H. Abdulla. *Principles of Psychoacoustics*, pages 15–49. Springer International Publishing, Cham, 2015. ISBN 978-3-319-07974-5. doi: 10.1007/978-3-319-07974-5_2. URL https://doi.org/10.1007/978-3-319-07974-5_2.

[92] D. Griffin and Jae Lim. Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243, 1984. doi: 10.1109/TASSP.1984.1164317.

[93] Tavish Vaidya, Yuankai Zhang, Micah Sherr, and Clay Shields. Cocaine noodles: Exploiting the gap between human and machine speech recognition. In *9th USENIX Workshop on Offensive Technologies (WOOT 15)*, Washington, D.C., August 2015. USENIX Association. URL https://www.usenix.org/conference/woot15/workshop-program/presentation/vaidya.

[94] Lea Schönherr, Katharina Kohls, Steffen Zeiler, Thorsten Holz, and Dorothea Kolossa. Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding, 2018.

[95] Lea Schönherr, Thorsten Eisenhofer, Steffen Zeiler, Thorsten Holz, and Dorothea Kolossa. Imperio: Robust over-the-air adversarial examples for automatic speech recognition systems, 2020.

[96] Joseph Szurley and J. Zico Kolter. Perceptual based adversarial audio attacks, 2019.

[97] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. Dolphinattack. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, Oct 2017. doi: 10.1145/3133956.3134052. URL http://dx.doi.org/10.1145/3133956.3134052.

[98] Hadi Abdullah, Muhammad Sajidur Rahman, Christian Peeters, Cassidy Gibson, Washington Garcia, Vincent Bindschaedler, Thomas Shrimpton, and Patrick Traynor. Beyond l_p clipping: Equalization based psychoacoustic attacks against ASRs. In Vineeth N. Balasubramanian and Ivor Tsang, editors, *Proceedings of The 13th Asian Conference on Machine Learning*, volume 157 of *Proceedings of Machine Learning Research*, pages 672–688. PMLR, 17–19 Nov 2021. URL https://proceedings.mlr.press/v157/abdullah21a.html.

[99] Zhuolin Yang, Bo Li, Pin-Yu Chen, and Dawn Song. Characterizing audio adversarial examples using temporal dependency. *CoRR*, abs/1809.10875, 2018. URL http://arxiv.org/abs/1809.10875.

[100] Cristina España-Bonet and José A. R. Fonollosa. Automatic speech recognition with deep neural networks for impaired speech. In Alberto Abad, Alfonso Ortega, António Teixeira, Carmen García Mateo, Carlos D. Martínez Hinarejos, Fernando Perdigão, Fernando Batista, and Nuno Mamede, editors, *Advances in Speech and Language Technologies for Iberian Languages*, pages 97–107, Cham, 2016. Springer International Publishing. ISBN 978-3-319-49169-1.

[101] Jean-Paul Haton. Automatic speech recognition: A review. In Olivier Camp, Joaquim B. L. Filipe, Slimane Hammoudi, and Mario Piattini, editors, *Enterprise Information Systems V*, pages 6–11, Dordrecht, 2005. Springer Netherlands. ISBN 978-1-4020-2673-7.

[102] Antony W Rix, John G Beerends, Michael P Hollier, and Andries P Hekstra. Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs. In *2001 IEEE international conference on acoustics, speech, and signal processing. Proceedings (Cat. No. 01CH37221)*, volume 2, pages 749–752. IEEE, 2001.

[103] Razvan Beuran, Mihail Ivanovici, and Bob Dobinson. Ntwork quality of service measurement system for application requirements evaluation. In *International Symposium on Performance Evaluation of Computer and Telecommunication Systems, SPECTS'03*, pages 380–387, 2003.

[104] Alexandros Kastanos, Anton Ragni, and Mark JF Gales. Confidence estimation for black box automatic speech recognition systems using lattice recurrent neural networks. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6329–6333. IEEE, 2020.

[105] Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael A. Specter, and Lalana Kagal. Explaining explanations: An approach to evaluating interpretability of machine learning. *CoRR*, abs/1806.00069, 2018. URL `http://arxiv.org/abs/1806.00069`.

[106] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6:52138–52160, 2018. doi: 10.1109/ACCESS.2018.2870052.

[107] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1), 2021. ISSN 1099-4300. doi: 10.3390/e23010018. URL `https://www.mdpi.com/1099-4300/23/1/18`.

[108] Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. A survey of the state of explainable AI for natural language processing. *CoRR*, abs/2010.00711, 2020. URL `https://arxiv.org/abs/2010.00711`.

[109] Xiaoliang Wu, Chau Luu, Peter Bell, and Ajitha Rajan. Explainable attribute-based speaker verification, 2024. URL `https://arxiv.org/abs/2405.19796`.

[110] Imen Ben-Amor, Jean-François Bonastre, Benjamin O'Brien, and Pierre-Michel Bousquet. Describing the phonetics in the underlying speech attributes for deep and interpretable speaker recognition. In *INTERSPEECH 2023*, pages 3207–3211, 2023. doi: 10.21437/Interspeech.2023-1648.

[111] Andrés Carofilis, Enrique Alegre, Eduardo Fidalgo, and Laura Fernández-Robles. Improvement of accent classification models through grad-transfer from spectrograms and gradient-weighted class activation mapping. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[112] Satvik Dixit, Daniel M. Low, Gasser Elbanna, Fabio Catania, and Satrajit S. Ghosh. Explaining deep learning embeddings for speech emotion recognition by predicting interpretable acoustic features, 2024. URL `https://arxiv.org/abs/2409.09511`.

[113] Tae-Wan Kim and Keun-Chang Kwak. Speech emotion recognition using deep learning transfer models and explainable techniques. *Applied Sciences*, 14(4):1553, 2024.

[114] Joan L Imbwaga, Nagaratna B Chittaragi, and Shashidhar G Koolagudi. Explainable hate speech detection using lime. *International Journal of Speech Technology*, 27(3): 793–815, 2024.

[115] Dima Shulga, Vered Silber-Varod, Diamanta Benson-Karai, Ofer Levi, Elad Vashdi, and Anat Lerner. Toward explainable automatic classification of children's speech disorders. In *International Conference on Speech and Computer*, pages 509–519. Springer, 2020.

[116] Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models. *URL https://openaipublic. blob. core. windows. net/neuron-explainer/paper/index. html.(Date accessed: 14.05. 2023)*, 2, 2023.

[117] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

[118] Andreas Krug, René Knaebel, and Sebastian Stober. Neuron activation profiles for interpreting convolutional speech recognition models. 2018.

[119] Gasper Begus and Alan Zhou. Interpreting intermediate convolutional layers of cnns trained on raw speech. *CoRR*, abs/2104.09489, 2021. URL `https://arxiv.org/abs/2104.09489`.

[120] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084, 2019. URL `http://arxiv.org/abs/1908.10084`.

[121] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with BERT. *CoRR*, abs/1904.09675, 2019. URL `http://arxiv.org/abs/1904.09675`.

[122] Devshree Patel, Param Raval, Ratnam Parikh, and Yesha Shastri. Comparative study of machine learning models and BERT on squad. *CoRR*, abs/2005.11313, 2020. URL https://arxiv.org/abs/2005.11313.

[123] Marko Robnik-Šikonja and Marko Bohanec. *Perturbation-Based Explanations of Prediction Models*, pages 159–175. Springer International Publishing, Cham, 2018. ISBN 978-3-319-90403-0. doi: 10.1007/978-3-319-90403-0_9. URL https://doi.org/10.1007/978-3-319-90403-0_9.

[124] John W Tukey et al. *Exploratory data analysis*, volume 2. Reading, MA, 1977.

[125] Amy Rafferty, Rudolf Nenutil, and Ajitha Rajan. Explainable artificial intelligence for breast tumour classification: Helpful or harmful. In *Interpretability of Machine Intelligence in Medical Image Computing: 5th International Workshop, iMIMIC 2022, Held in Conjunction with MICCAI 2022, Singapore, Singapore, September 22, 2022, Proceedings*, pages 104–123. Springer, 2022.

[126] N Arun, N Gaw, P Singh, K Chang, M Aggarwal, B Chen, et al. Assessing the (un) trustworthiness of saliency maps for localizing abnormalities in medical imaging. arxiv. *arXiv preprint arXiv:2008.02766*, 2020.

[127] Yi-Shan Lin, Wen-Chuan Lee, and Z. Berkay Celik. What do you see? evaluation of explainable artificial intelligence (XAI) interpretability through neural backdoors. *CoRR*, abs/2009.10639, 2020. URL https://arxiv.org/abs/2009.10639.

[128] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10):1084–1102, 2018.

[129] Mengjiao Yang and Been Kim. Benchmarking attribution methods with relative feature importance. *arXiv preprint arXiv:1907.09701*, 2019.

[130] Yilun Zhou, Serena Booth, Marco Tulio Ribeiro, and Julie Shah. Do feature attribution methods correctly attribute features? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 9623–9633, 2022.

[131] Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. *Advances in neural information processing systems*, 32, 2019.

[132] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*, 2018.

[133] Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Bach, and Klaus-Robert Müller. Evaluating the visualization of what a deep neural network has learned. *CoRR*, abs/1509.06321, 2015. URL http://arxiv.org/abs/1509.06321.

[134] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Suggala, David I Inouye, and Pradeep K Ravikumar. On the (in) fidelity and sensitivity of explanations. *Advances in Neural Information Processing Systems*, 32, 2019.

[135] Zhong Qiu Lin, Mohammad Javad Shafiee, Stanislav Bochkarev, Michael St Jules, Xiao Yu Wang, and Alexander Wong. Do explanations reflect decisions? a machine-centric strategy to quantify the performance of explainability algorithms. *arXiv preprint arXiv:1910.07387*, 2019.

[136] Michael Jessen. *Speaker Classification in Forensic Phonetics and Acoustics*, pages 180–204. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. ISBN 978-3-540-74200-5. doi: 10.1007/978-3-540-74200-5_10. URL https://doi.org/10.1007/978-3-540-74200-5_10.

[137] John H.L. Hansen and Taufiq Hasan. Speaker recognition by machines and humans: A tutorial review. *IEEE Signal Processing Magazine*, 32(6):74–99, 2015. doi: 10.1109/MSP.2015.2462851.

[138] Raghavendra Pappagari, Tianzi Wang, Jesus Villalba, Nanxin Chen, and Najim Dehak. x-vectors meet emotions: A study on dependencies between emotion and speaker recognition, 2020. URL https://arxiv.org/abs/2002.05039.

[139] Jennifer Williams and Simon King. Disentangling style factors from speaker representations. pages 3945–3949, 09 2019. doi: 10.21437/Interspeech.2019-1769.

[140] Desh Raj, David Snyder, Daniel Povey, and Sanjeev Khudanpur. Probing the information encoded in x-vectors. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, December 2019. doi: 10.1109/asru46091.2019.9003979. URL http://dx.doi.org/10.1109/ASRU46091.2019.9003979.

[141] Soumi Maiti Erik Marchi Alistair Conkie. Generating multilingual voices using speaker space translation based on bilingual speaker data. In *ICASSP*, 2020. URL https://arxiv.org/pdf/2004.04972.pdf.

[142] Chau Luu, Peter Bell, and Steve Renals. Leveraging speaker attribute information using multi task learning for speaker verification and diarization. *CoRR*, abs/2010.14269, 2020. URL https://arxiv.org/abs/2010.14269.

[143] Wiebke Toussaint Hutiri and Aaron Yi Ding. Bias in automated speaker recognition. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, page 230–247, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393522. doi: 10.1145/3531146.3533089. URL https://doi.org/10.1145/3531146.3533089.

[144] Mariel Estevez and Luciana Ferrer. Study on the fairness of speaker verification systems across accent and gender groups. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[145] Mariel Estevez and Luciana Ferrer. Study on the fairness of speaker verification systems on underrepresented accents in english. *arXiv preprint arXiv:2204.12649*, 2022.